Predict patient deterioration in hospital's general ward

Vesa Ikonen

Master's Thesis

Big Data Analytics

2019

MASTER'S THESIS	
Arcada	
Degree Programme:	Big Data Analytics
Identification number:	7618
Author:	Vesa Ikonen
Title:	Predict patient deterioration in hospital's general ward
Supervisor (Arcada):	Anton Akusok
Commissioned by:	GE Healthcare Finland Oy

Abstract:

Increasing amount of patient monitoring data is available in hospitals in an electronic format. Patient data is mostly available from hospital departments which provide intensive treatment, but wireless and wearable sensors will enable constant monitoring of vital sign parameters also in hospitals general ward. High-frequency real-time patient data enables development of systems which automatically notify clinicians if patient condition deteriorates. The situation can be even further improved by developing data analytics tools which are able to predict changes in patient condition.

The aim of this study is to develop machine learning tools for predicting patient deterioration in hospital's general ward which supports continuous monitoring of vital sign parameters. The novel idea of the study is to simulate general ward data by selecting a subset of the intensive care unit data. Patient deterioration is defined by National Early Warning Score (NEWS) threshold of 7. The task is to predict if patient deteriorates within 2 hours prediction window using a gap of one hour between the prediction time and prediction window. Patient's medical information from the past one hour is utilized in the prediction. Four supervised machine learning models are trained for the classification; logistic regression, kernelized support vector machine, random forest and gradient boosting classifier. The best model is searched using a grid search with 10-fold cross-validation in the development set.

Gradient boosting classifier achieves the best cross-validation performance; AUROC of 0,813 and AUPRC of 0,375. In the test set evaluation, model's AUROC is 0,808 and AUPRC is 0,348. Test set prevalence is 10,8%. Using the model's default threshold, test set sensitivity is 0,744 and precision is 0,240. It means that the model correctly detects 3 out of 4 deteriorations. Among all the predicted deteriorations, the proportion of correct predictions is 1 in 4.

Gradient boosting classifier's test set performance is compared to NEWS scoring system's medium level clinical alert which works as a baseline. Using model's threshold which provides the same sensitivity as baseline, gradient boosting classifier has 25% less false positives. Using model's threshold which provides the same precision as baseline, gradient boosting classifier has 45 % higher sensitivity than the baseline.

The achieved results suggest that real-time prediction of patient deterioration based on the NEWS could assist clinicians in identifying deteriorating patients in hospitals general ward. NEWS is widely used in hospitals general ward and providing NEWS prediction in addition to real-time NEWS could help clinical staff in focusing in most critical patients. The achieved precision remains moderate in the study and could be a topic for the future research.

Keywords:	Machine learning, gradient boosting classifier, healthcare, general ward, NEWS
Number of pages:	78
Language:	English
Date of acceptance:	03.03.2020

CONTENTS

Figures

Tables

1 INTRODUCTION

1.1 Background

Patient's condition in hospital is monitored by measuring physiological parameters from the body. Measured parameter values help clinicians to assess the medical state of the patient. Number of measured parameters vary depending on the condition of the patient and the department of the hospital. Most important physiological parameters, so called vital signs, include body temperature, heart rate or pulse rate, respiratory (breathing) rate and blood pressure. Oxygen saturation of arterial blood is often used as a fifth vital sign. Vital signs indicate the state of basic body functions. In addition to vital sign parameters, numerous additional parameters may also be measured depending on the treatment needs and medical state of the patient.

During the hospital visit, patient is located in one of the hospital's departments depending on the treatment needs. The intensive care unit (ICU) provides intensive treatment for patients who have severe illness or injury. In ICU, patient is closely monitored using patient monitoring devices which constantly measure several physiological parameters. Operating room (OR) is another example of the department where patient is closely monitored. During the anesthesia in OR, minimum set of measured parameters is determined by medical standards. In general ward, on the other hand, patients are usually in a better condition and monitoring is less intensive. In general ward, patient is typically not connected to any constant monitoring device, but patient's vital sign parameters are measured at defined intervals by hospital staff. Also, fewer parameters are measured in general ward.

Patient monitoring in hospital varies in many ways: how many parameters are measured, how often values are measured and if measurements are done automatically by a device or manually by hospital staff. In all cases, patient measurement data is nowadays stored in digital format in an electronic medical record (EMR). EMR stores patient's medical state over a time. If patient is constantly monitored by a device, a snapshot of physiological parameters is frequently taken and transferred to the EMR. Depending on

the hospital system, measured physiological parameter values may be automatically transferred from the measuring device to the EMR system or they may be manually entered by clinicians. In addition to measured physiological parameter values, the EMR contains also other medical information like medicine data, laboratory test results, images, textual notes and information about illnesses. Also, demographic data like age and gender are included in the EMR.

Patients' constant monitoring is becoming more and more common. In addition to departments like OR and ICU which require intensive monitoring, also other departments like general ward may nowadays sometimes have monitoring devices that constantly measure patient's physiological parameters. Because patients in the general ward are normally in a better condition than e.g. in the ICU, the requirements for the monitoring devices are different. In the general ward, patient may for example move independently which means that patient cannot be attached to a stationary monitoring device like in the ICU or OR.

The development in the medical field is going towards wearable and wireless patient monitoring devices that measure physiological parameters but allow also patient movement. In the future, wearable sensors will enable constant monitoring more often also in the general ward but the number of monitored parameters will probably be less than e.g. in the ICU. In the future, patient stay at the hospital can also be shortened if patient can be monitored remotely from the home. This technological development means that there will be more patient data available in the future and also more data available from other environments than those hospital departments that provide intensive patient treatment.

1.2 The problem

Patient monitoring helps clinicians in assessing the medical state of the patient. In ICU, patients are in a life-threatening condition and they are intensively monitored by devices and clinical experts. Constant monitoring of physiological parameters and physical presence of clinicians help hospital staff to react fast to changes in patient condition. In general ward, the situation is different. When patient is moved from the ICU or OR to the general ward, the level of monitoring decreases. In the general ward, fewer parame-

ters are measured, values are measured less frequently and there are more patients for one clinician to take care of. Patient's vital signs may be measured e.g. once every six hours by a nurse who visits the patient's bed. If patient condition deteriorates between the visits, it may be unnoticed.

Sometimes patient condition may change rapidly. For example, sepsis which is one of the main causes of death in hospitals, is difficult to diagnose. When the signs of sepsis are visible, patient may already have high risk for mortality. It is thus important to detect sepsis early and start treatment as soon as possible. The sooner the diagnosis is done the better is the treatment outcome. Early identification and fast clinical intervention can decrease patient's mortality caused by sepsis (Torsvik et al. 2016). Cardiac arrest is another example of clinical deterioration that can occur suddenly. For patients having a coronary disease, cardiac arrest can occur as an unanticipated sudden event (Smith et al. 2013 p. 8). For the successful treatment outcome, early recognition of deteriorating patients and fast clinical responses are essential.

The current convention in general wards is to use a scoring system to detect deteriorating patients and to determine how often patient's vital signs should be measured. A scoring system takes one or more measured physiological parameter values, compares measured values against normal value ranges and uses set of rules to produce one number as an output. Produced output score helps clinicians to assess patient's medical condition and it helps to identify patients who require more intensive treatment and monitoring. An example of multiparameter scoring systems is Early Warning Score (EWS). EWS system is not standardized but multiple variants of EWS are in use depending on the hospital and country. A widely used EWS system is National Early Warning Score (NEWS) which was developed in the UK by Royal College of Physicians (2017). The NEWS scoring system was published 2012 and was updated 2017. The purpose of the development was to standardise early warning system across the NHS (National Health Service, UK) but NEWS is also widely used outside the UK. NEWS utilizes the following physiological parameters for producing the output score: respiration rate, oxygen saturation, knowledge if supplemental oxygen is given, body temperature, systolic blood pressure, heart rate and level of consciousness. For each parameter, a subscore is calculated and the total NEWS score is a sum of all subscores. Produced output score

describes the risk level of a patient. When the output score increases, patient should be monitored more frequently. In addition to score calculation rules, the NEWS scoring system also gives recommendations for clinical alert thresholds and clinical responses to them. According to Royal College of Physicians (2017), a NEWS value of 7 should trigger a high-level clinical alert and the response is usually to transfer the patient to the ICU to the continuous monitoring. A NEWS value of 5 is a threshold for a medium level alert which is an indication of potential clinical deterioration. The response to medium level alert is urgent clinical assessment.

A limitation of NEWS and other similar kind of scoring systems is that they do not take into consideration patient's personal parameter baselines but instead use fixed ranges for calculating parameter subscores. Another limitation is that the score is typically calculated using only the latest available physiological parameter values and parameters' temporal changes are not taken into account. Despite their limitations, scoring systems are widely used at hospitals to assist clinical staff in identifying deteriorating patients. There are several devices in the market that automatically calculate EWS or some other similar kind of index based on the measured parameter values. There is also some evidence available that EWS systems can improve patient outcomes at hospitals although their efficiency in practice highly depends on the patient cohort, used threshold scores, implementation of appropriate responses, available hospital resources and clinical staff (Le Lagadec & Dwyer 2017). In the general ward, the efficiency of scoring system also depends on how often the patient is monitored. In the general ward, patient's physiological parameters are measured infrequently and the score may be calculated e.g. once every six hours when the nurse visits the patient's bed.

Substantial amount of patient deteriorations happens in hospitals' general ward. The problem in the general ward is that patient deterioration may be unnoticed because of lack of constant monitoring. Early detection of deterioration and fast clinical intervention is crucial for the successful treatment outcome. Wireless wearable monitoring devices will enable earlier clinical interventions in general wards. Constant monitoring of physiological parameters together with the capability of remotely alarm clinicians will shorten the reaction time to changes in patient condition. The situation can be even further improved by developing devices that do not only react to patient's current situation

but are able to predict changes in patient condition. Predictions will give clinicians even more time to prepare proper treatment for the patients. Constant monitoring will enable also this development because devices provide more measurement data that can be used in data analysis.

1.3 Aim of the study

The aim of this study is to develop machine learning tools for predicting patients' deterioration in the hospital's general ward. The target environment is a general ward which supports constant monitoring of patient's physiological parameters. Other EMR data like laboratory data, drug data and demographics data will also be utilized in the prediction.

This study is motivated by the following overall efforts of patient care:
- prevent patient's permanent injury or death
- shorten patient's stay at the hospital
- prevent patient admissions to ICU. ICU treatment is more expensive and hospital's ICU capacity is more limited.

From the clinical point of view, deterioration in this thesis means that patient's medical condition becomes worse from any reason. The deterioration may mean for example sepsis, cardiac (heart) failure, respiratory (breathing) failure or some other organ dysfunction. The attempt to predict patient deterioration is based on the research knowledge that 85 % of severe adverse events are preceded by abnormal behavior of physiological parameters (Le Lagadec & Dwyer 2017 p. 1). Abnormal behavior of physiological parameters could be detected by machine learning tools.

The thesis is done for the GE Healthcare Finland Oy.

1.4 Dataset

This chapter gives an overview of data that is utilized in this thesis. More detailed description about the dataset and how it is processed is given by the chapter 3.1. Data that is utilized in this study is collected from the ICU department of HUS Jorvi hospital.

ICU data is used because constant monitoring devices are not yet common in the general ward and that's why real data from the general ward is not available. The purpose of the study is to select a subset of ICU data that simulates general ward and to use that subset as a final dataset of the study. Defining which part of ICU data simulates general ward and to extract that data from the original ICU dataset forms a significant part of this study.

The ICU dataset is extracted from the hospital's EMR system. The dataset contains physiological parameter values that have been measured from the patients at the hospital. Other EMR data like medication data, patient demographics and knowledge about used medical devices is also included in the dataset. The dataset contains de-identified data.

The dataset is not pre-labeled but labels must be created as a part of this thesis. For each moment of time the dataset must have a label which describes the medical condition of a patient. The labeling requires that patient medical condition can be expressed as a function of physiological parameter values that are available in the dataset. This is how currently used scoring systems, like ESW, work. In this study, the NEWS is used to represent the medical condition of a patient.

1.5 Research questions

The purpose of this thesis is to answer to the following questions:

1. How well patients' deterioration can be predicted using the simulated ward dataset?
2. What kind of dataset can be created by simulating general ward data from ICU data? Most important questions are what is the size and prevalence of the dataset and the number of patients in the dataset?
3. Which machine learning model has the best performance in predicting patient deterioration?

1.6 Limitations

The purpose of this thesis is to create a proof of concept and prototype implementation. The production level implementation is out of the scope of this thesis. Also, the description about the final production environment is not in the scope of this thesis.

1.7 Abbreviations

AUPRC	Area Under Precision-Recall Curve
AUROC	Area Under Receiver Operating Characteristics Curve
Cr	Creatinine
CV	Cross-Validation
DiaBP	Diastolic Blood Pressure
ED	Emergency Department
EMR	Electronic Medical Record
EWS	Early Warning Score
FiO2	Fraction of Inspired Oxygen
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
GCS	Glasgow Coma Scale
HCT	Blood Hematocrit (volume percentage of red cells in blood)
HR	Heart Rate
ICU	Intensive Care Unit.
K	Potassium (Kalium)
LR+	Positive Likelihood Ratio
LR-	Negative Likelihood Ratio
MeanBP	Mean Blood Pressure
MEWS	Modified Early Warning Score
Na	Sodium (Natrium)
NEWS	National Early Warning Score
OR	Operating Room

PPV	Positive Predictive Value
qSOFA	Quick Sequential Organ Failure Assessment
RR	Respiration Rate
RTT	Rapid Response Team
SIRS	Systemic Inflammatory Response Syndrome
SpO2	Oxygen saturation of arterial blood
SysBP	Systolic Blood Pressure
Temp	Temperature
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
WBC	White Blood Cell Count (Leukocyte Count)

2 LITERATURE REVIEW

2.1 Overview

This chapter presents an overview of research papers related to the patient deterioration in hospitals. The first chapter examines studies related to the deterioration and how well currently used scoring systems can detect deteriorating patients. The second chapter focuses on predictive tools that have been developed to identify deteriorating patients.

2.2 Patient deterioration

A significant number of patients deteriorate on hospitals' general ward. An international study was conducted in 2011 to determine postoperative mortality among patients that underwent non-cardiac surgery. The study collected data from 498 hospitals across 28 European countries. The total number of included patients in the study was 46 539. The study reported that 1864 (4%) patients died during their hospital visit. A notable observation was that: "1358 (73%) patients who died were not admitted to critical care at any stage after the surgery". Furthermore, 43% of those patients that died after admission to critical care, died after they had been transferred from critical care to general ward. (Pearse et al. 2012)

Early detection of deterioration and fast clinical intervention are crucial for successful treatment outcome. These facts are underlined by the following two studies. Cardoso et al. (2011) evaluated how much a delay in admission from general ward to ICU affects patient mortality. In their study they reported that each hour of delay in admission to ICU was associated with 1,5% increase in risk of ICU death and 1% increase in hospital death.

One of them main causes of death in hospitals is sepsis. It is difficult to diagnose and undetected or untreated sepsis can quickly progress to septic shock where blood pressure drops to dangerously low level. Kumar et al. (2006) found out that every hour of

delay in administration of antibiotics increases the mortality by 7,6% for sepsis shock patients with hypotension (low blood pressure).

The current convention in hospitals general ward is to use a scoring method for identifying deteriorating patients and to determine how often patients should be monitored. Churpek et al. (2017) compared widely used scoring methods qSOFA, SIRS, NEWS and MEWS for detecting deteriorating patients outside the intensive care unit. qSOFA and SIRS are scores which focus on sepsis detection while NEWS and MEWS are more general scores trying to identify deteriorating patients for any clinical reason. They studied how well these methods predict death and ICU transfer. 30 677 patients who were suspected to have an infection outside the ICU were included in the study. They calculated patient's highest score from the stay outside of the ICU and evaluated how well it predicted in-hospital mortality and ICU transfer. Their conclusion was that, among the studied scores, NEWS was the most accurate score for predicting in hospital mortality (area under the receiver operating characteristics curve, AUROC, 0,77) followed by MEWS (AUROC 0,73). For the combined outcome, death or ICU transfer, results were similar but AUROCs slightly lower.

Smith et al. (2013) conducted a study where they compared the performance of NEWS against 33 other early warning scores. The study tested how well scores are able to discriminate patients at risk of cardiac arrest (CA), unanticipated ICU transfer or death within 24 hours. Their dataset consisted of 35 585 acute medical admissions. Their performance evaluation was based on the AUROC values. They concluded that NEWS is better in discriminating patients at risk of the combined outcome of CA, unanticipated ICU transfer or death than other tested 33 scores. NEWS performance was also better for individual outcomes of unanticipated ICU transfer and death, but not for CA. They discuss that it may be because “cardiac arrest is less predictable, sometimes occurring as an unanticipated, sudden event, occurring in a patient with coronary disease in the absence of antecedent physiological disturbance.” They continue that this is in contrast to unanticipated ICU transfer and death which are almost always preceded by deranged physiological parameters. They reported AUROCs for NEWS 0,722/0,857/0,894/0,873 for outcomes CA/unanticipated ICU transfer/death/combined outcome.

2.3 Predicting patient deterioration

Numerous studies have developed tools for predicting patient deterioration. The implemented studies vary in many ways: what are the input parameters, what is the predicted outcome, how far in the future the outcome is predicted, how much history data (if any) is utilized in the prediction and what is the target population (hospital department). Also, metrics that are used to evaluate the performance may differ. Many investigated studies focus on ICU environment, apparently because there is more data is available from the intensive care departments. Continuous monitoring of several physiological parameters provides datasets that include more parameters and higher measurement frequencies compared to other hospital departments. Therefore, models that have been developed for the ICU are better able to utilize trend information from the parameters' time series data. Tools that have been developed for the general wards typically use discrete measurement values that are calculated from a time window of some hours in length. The rest of this chapter describes those examined studies that are most relevant to this thesis. A direct reference to this study was not found because the aim of this work is to predict patient deterioration in the (future) general ward environment which supports continuous monitoring of patient's physiological parameters.

Churpek et al. (2012) developed a cardiac arrest risk triage (CART) score to predict cardiac arrest (CA) using vital signs available in general wards. The total number of patients included in the study was 47 272. Their model utilized the following data from the EMR: respiratory rate, heart rate, diastolic blood pressure and age. Measurements within 30 minutes of CA were excluded in order to provide enough time for the clinical intervention. They compared their CART score to widely used Modified Early Warning Score (MEWS) and reported that CART score predicted CA better than MEWS (AUROC for CART: 0,84, AUROC for MEWS: 0,76). Using the specificity of 89,9%, sensitivity for CART was 53,4%, sensitivity for MEWS was 47,7%. Additionally, they also compared scores' ability to predict ICU transfer (AUROC for CART: 0,71, AUROC for MEWS: 0,67). In this study, scores were calculated for eight hours’ time window, using vital sign values closest to the beginning of the time window. Scores were calculated for 48 hours before the event.

Later Churpek et al. (2016) conducted a study where they compared the performance of several machine learning methods and modified early warning score (MEWS) in predicting patient deterioration on hospital medical-surgical wards. In their study they developed 9 machine learning models that predicted the combined outcome of cardiac arrest, ICU transfer and death. I.e. they considered their problem as a binary classification problem. Developed models utilized physiological parameters, laboratory data and demographics data of 269 999 patients. They used discrete 8-hour time windows because of the available measurement frequency of physiological parameters. Parameter values closest to the beginning of 8 hours' time window were used as a model input and as an output the model predicted if the outcome occurred within the time window. According to their report, all machine learning methods performed better than MEWS. Their best model was a random forest with AUROC 0.80 while MEWS had AUROC 0.7. They reported that MEWS AUROC was "determined using whether an event occurred within twenty-four hours of each individual observation because this is a standard metric for early warning score comparisons". Respiratory rate and heart rate were evaluated to be the most important predictor variables for the random forest model. As a conclusion they said that investigated techniques "may result in improved identification of critically ill patients on the wards".

Alvarez et al. (2013) used EMR data of 7466 patients to create a logistic regression model for predicting the following combined outcome: out of ICU cardiac arrest, acute respiratory compromise and unexpected death. They included patients that were admitted to the internal medicine ward from either the emergency department (ED) or outpatient clinics. Additionally, patients that were admitted to the ICU from the ED were included. In ICU, outcome events cardiac arrest, acute respiratory compromise and unexpected death, were included if they occurred within the first 24 hours of ICU stay. Model features were collected from the previous 24 hours and included age, SpO2, diastolic blood pressure, arterial blood gas and laboratory values, emergent orders, and assignment to a high-risk floor. Data within one hour of an event was excluded. They compared the performance of their model to the Modified Early Warning Score (MEWS) and human judgement activated rapid response team (RTT). They reported AUROC of 0,85 for their model and AUROC of 0,75 for the MEWS. Other reported model metrics were sensitivity 0,516, specificity 0,943 and precision 0,10. When comparing to institu-

tional RTT deployment, their model had better sensitivity than the RTT (model: 0,516, RTT: 0,258), but RTT had better precision (model: 0,10, RTT: 0,21). They reported that developed model was able to identify patients at risk several hours before the RTT.

Wu et al. (2017) developed L2 regularized logistic regression classifier for predicting patient's need for vasopressor administration and patient's vasopressor weaning readiness in ICU. Vasopressors are powerfull drugs that are used to elevate blood pressure of critically ill patients. They used publicly available MIMIC II database which contains ICU data of 26 870 hospital admissions. Patients which had less than 12 hours or more than 96 hours of data were excluded in order to avoid sicker patients. Also, they predicted only the patient's first vasopressor administration, because patients with multiple administrations can be in a different physiological condition. They included 19 variables from the dataset: 6 physiological parameters, 4 laboratory measurements and 7 static variables. Physiological data was hourly sampled. Static variables included SAPS I and SOFA scores. They utilized four hours of history data in the prediction. They used discretization and binning of features. The task was to predict patient's need for vasopressor medication within the 2 hours' time window both without gap and with 4 hours gap (between prediction time and target window). In addition, they predicted patient's vasopressor weaning readiness within the next 2 hours without a gap. They reported AUROCs of 0,92, 0,88 and 0,71 for vasopressor administration without gap, vasopressor administration with 4 hours gap and vasopressor weaning readiness.

Earlier, Fialho et al. (2013) had also predicted vasopressor administration need in ICU. They used 2944 patients from the MIMIC II dataset who needed fluid resuscitation and among these patients they predicted if patients will require vasopressors or not. They predicted patient's vasopressor need within the next two hours (without a gap) using two disease-based models and one general model. They reported AUROC of 0,79 for their general model and AUROCs of 0,82 and 0,83 for pneumonia and pancreatis disease-based models. Their conclusion was that model performance is better with disease-based models which use only a subset of patients (patients who have diagnosed pneumonia or pancreatitis).

Suresh et al. (2017) developed neural network-based models for predicting multiple clinical interventions in ICU. Prediction targets included e.g. invasive ventilation, non-invasive ventilation and vasopressors. Ventilation means hospital treatment that assists patient in breathing. They used general ICU population unlike Fialho et al. (2013) who used a subset of patients receiving fluid resuscitation. In their work, Suresh et al. used 6 hours history window, 6 hours gap and 4 hours prediction window. They used MIMIC III database and included patients who had ICU stays between 12 and 240 hours. They reported AUROC of 0,75 for predicting a need for invasive ventilation, AUROC of 0,76 for predicting a need for non-invasive ventilation and an AUROC of 0,77 for predicting a need for vasopressors.

Ghassemi et al. (2017) used also MIMIC III database for predicting clinical interventions, like mechanical ventilation and vasopressor administration, in ICU. Patients with less than 6 hours or more than 360 hours of data were excluded. Their final dataset was 36 050 patients. They trained a separate binary classifier for each target. They used hourly sampled data and evaluated performance for 1, 2, 4 and 8 gap hours. At each one-hour interval, they predicted whether to apply the intervention or not. Data until the first positive intervention was used. Their AUROCs for vasopressor administration were 0,82/0,81/0,78/0,74 for gaps of 1h/2h/4h/8h. AUROCs for ventilation were 0,68/0,68/0,67/0,66 for gaps of 1h/2h/4h/8h.

Calvert et al. (2016) used ICU dataset to develop a sepsis prediction method using nine commonly available physiological parameters: systolic blood pressure, pulse pressure, heart rate, temperature, respiration rate, white blood cell count, pH, blood oxygen saturation and age. As an output their method calculated the risk that a patient will have a sepsis. They divided the patient's hospital stay into one-hour time windows and rounded up measurement times to the nearest hour. For producing the output score they developed an equation which utilized physiological parameters, parameters' trend information and also trend information among pairs and triplets of parameters. They reported average area under ROC curve of 0,83 across predictive times up to three hours before the onset of sepsis. According to the report their results exceed or rival with existing biomarker detection methods. They conclude that their model's "key feature is the ability

to combine diverse measurements and find correlations of these aggregate measurements with patient outcomes of interest".

2.4 Conclusions

The purpose of this thesis is to predict patient deterioration in the general ward for any clinical reason. The NEWS score is selected to represent patient's medical condition and to define the deterioration. No other study was found that would have predicted the same target. Most common targets when predicting patient deterioration outside the ICU were cardiac arrest, unplanned ICU transfer and death. With ICU data, targets like sepsis, mechanical ventilation or vasopressor administration were also used. These targets are probably easier to define because their occurrence is recorded in the dataset. They may also be more accurate than a score which a surrogate of patient's medical state and may sometimes be inaccurate for example because of measurement artifacts. It might also be that generic deterioration is too abstract thing to predict successfully although e.g. combination of cardiac arrest, unplanned ICU transfer and death can also be understood to represent generic deterioration.

In this study, the dataset is manually labeled so that it contains patient's medical condition. According to comparison of scores, widely used National Early Warning Score (NEWS) seems to be a good candidate for determining the medical state of the patient. NEWS does not focus on any specific medical condition but tries to identify patient deterioration for any clinical reason. In the reviewed studies, NEWS was found to have better performance than compared scores.

In order to be clinically meaningful, the prediction of adverse event should happen early enough to give time for the clinical intervention (change in patient treatment, medication, patient transfer to another department etc.). Predicting patient deterioration e.g. some minutes before the onset is not useful in practice. Instead, prediction should likely happen at least an hour in advance in order to be effective in real world hospital environment.

Wide range of machine learning methods have been tried in medical domain. Based on this limited literature review no single method was found to be superior to the others. Performance results vary depending on the test setup. Linear models, random forests and neural networks were most often referred in the examined papers, but it was not possible to make any conclusions for this thesis.

Metrics that were most often used in evaluating the performance of models were sensitivity, specificity and area under the receiver-operating characteristic curve (AUROC). AUROC was clearly the main metric. These metrics should be used also in this study in order to provide somewhat comparable results. A shortcoming was that only some studies reported precision, which is another interesting metric in the medical domain. High AUROC can be achieved even if the precision remains low. In the medical domain, alarm fatigue is a known problem and it is important to avoid false positives.

Reported model performances seem to depend highly on the test setup. Even if the prediction target remains the same, the reported performance varies depending on the test population, used window sizes, input parameters etc. It was thus not possible to find a clear baseline performance for this thesis.

3 RESEARCH METHODOLOGY

This chapter describes the dataset preparation process and machine learning methods that are applied in this study. The chapter is divided into two parts. The first part (chapter 3.1) describes how data that simulates general ward is created from the original ICU dataset. This part also shows statistics about the created dataset. The second part (chapter 3.2) describes the machine learning methods that are applied to the created dataset for predicting patient deterioration. The purpose of the whole work is to predict patient deterioration from any clinical reason using the simulated general ward data.

3.1 Dataset preparation

3.1.1 ICU data as a starting point

The purpose of the dataset preparation is to create a labeled dataset which simulates general ward data. The starting point is a dataset that is collected from HUS Jorvi hospital. The dataset contains data from patients that were admitted to hospital's intensive care unit (ICU) between July 2001 and December 2017. The size of the dataset is 6213 patient stays. A patient stay contains data of one patient from the ICU admit to the ICU discharge. If the same patient is admitted to the ICU several times, they are recorded as different patient stays in the dataset. The dataset is extracted from the hospital's EMR system. The dataset contains physiological parameter values that have been measured from the patients during their stay at the ICU. Vital sign parameters are continuously measured at fixed intervals. Other EMR data, like laboratory test results, medication data, knowledge about used medical devices and demographics data are also available in the dataset. Laboratory test results, medication and device information are intermittently recorded data, demographics data is static data. The dataset is an SQL database. Data is de-identified.

The dataset was provided as a database dump which was imported to the local PostgreSQL database. Local database was used to explore dataset and to develop SQL scripts for extracting data from the database. Dataset was also randomly split into

development set and test set in the SQL database so that development set contains 75% of patient stays and test set contains 25% of patient stays.

3.1.2 Patient stay candidates

The original SQL database contains 6213 ICU patient stays. However, only the following patient stays are considered by this study and extracted out from the database:

- stays where patient is 16 years or older. This study calculates NEWS score from the patient data and Royal College of Physicians (2017) recommends that NEWS should not be used for children that are younger than 16 years. Children are excluded because their physiological parameters' normal values differ from adults.
- stays that are longer than 24 hours. Patients are admitted to ICU when they have severe illness or injury. This study does not use data from the first 24 hours of ICU stay because patients' medical condition does not correspond to ward patients' typical medical condition.

4419 patient stays fulfilled the above mentioned criteria and were extracted from the SQL database for further processing. 3318 patient stays were extracted from the development set and 1101 were extracted from the test set. Extracted patient stays were stored in the csv files. Chapter 3.1.3. describes in detail the data that is extracted for each patient stay.

Note that Royal College of Physicians (2017) recommends also that NEWS should not be used for women who are pregnant. This recommendation is not followed in this study because pregnancy information was not available in the database.

3.1.3 Extracted patient stay data

This chapter describes patient stay data that is extracted from the original ICU database. The same information is extracted for all patient stays. Extracted data is later used for calculatating features and prediction targets and for selecting data periods that simulate general ward.

Vital sign parameters

The following physiological parameters are continuously measured at fixed time intervals in the dataset. Their measurement interval is around two minutes (see Table 4 for details). These parameters are extracted from the database and used for calculating model features and the NEWS score. Parameter set is selected so that the same parameters will be available in the target general ward. Values are extracted from the database only if values are within the range defined by Table 1:

Table 1. Continuously measured physiological parameters that are extracted from the database

Parameter	**Abbreviation**	**Unit**	**Range**
Respiration rate	RR	1/min	0 - 100
Heart rate	HR	1/min	20 - 200
Systolic blood pressure	SysBP	mmHg	10 - 300
Diastolic blood pressure	DiaBP	mmHg	0 - 200
Mean blood pressure	MeanBP	mmHg	0 - 200
Body temperature	Temp	°C	0 - 45
Oxygen saturation of arterial blood	SpO2	%	25 - 100

Laboratory test results

Laboratory test results are intermittent data (occasionally recorded). The database contains many different laboratory tests. The following subset is extracted in this study because these tests are typically done daily in the general ward. Laboratory test results are used as model features:

Table 2. Laboratory tests that are extracted from the database

Lab test	**Abbreviation**	**Unit**	**Range**
Creatinine	Cr	µmol/l	0 – 1500
Potassium (Kalium)	K	mmol/l	1 – 10
Sodium (Natrium)	Na	mmol/l	100 – 170
White blood cell count (leukocyte count)	WBC	E9/l	0 – 100
Blood hematocrit (volume percentage of red blood cells in blood)	HCT	%	10 – 80

Demographics

Demographic data is static data which does not change during the patient's hospital stay. The following demographics data is used as model features:

- Age (hospital admission age)
- Gender

Level of consciousness

Glasgow Coma Scale (GCS) is a scoring system which is used to assess patient's level of consciousness. The total GCS value is a sum of three subscores; patient's eye response, verbal response and motor response. The smallest possible GCS value is 3 which correspods to coma or death. The highest possible GCS value is 15 which means that patient is fully awake. GCS subscores are intermittently recorded in the dataset. They are extracted from the database and used as model features and for calculating the NEWS score.

Table 3. Glasgow Coma Scale elements that are extracted from the database

GCS element	**Abbreviation**	**Range**
GCS eye response	GCSe	1 – 4
GCS verbal response	GCSv	1 – 5
GCS motor response	GCSm	1 – 6

Supplemental oxygen

In supplemental oxygen therapy patient gets extra oxygen. Patient may get supplemental oxygen for example because of lung or heart problems. This information is not directly available in the database. Instead, knowledge if supplemental oxygen is given is derived from the patient's measured FiO2 value (fraction of inspired oxygen). If FiO2 value is > 21%, a patient is considered to have supplemental oxygen treatment. Normal air contains 21% of oxygen. Time periods where FiO2 value is <= 21% or FiO2 measurement value is not available are considered as not containing supplemental oxygen therapy.

Knowledge about supplemental oxygen treatment is used as a model feature and in NEWS score calculation.

If measured FiO2 value is > 50%, a patient is considered to get pressurized supplemental oxygen therapy. If measured FiO2 value is <= 50% or FiO2 measurement value is not available, patient is considered not to get pressurized supplemental oxygen. Pressurized supplemental oxygen therapy is not given in general ward and this information is used to select data periods that simulate general ward.

FiO2 parameter is only used to determine if supplemental oxygen or pressurized supplemental oxygen therapy is given. FiO2 data is not otherwise utilized in this study.

Vasopressor medication periods

Vasopressors are drugs that strongly affect the blood pressure. They elevate mean arterial pressure by creating vasoconstriction. Vasopressors are often prescribed for example for patients that have a septic shock (D'Aragon et al. 2015). Vasopressors are not administered in the general ward and knowledge about vasopressor medication is used to select data periods that simulate general ward. In this study, vasopressor medication equals to administration of any of the following medicines:

- Dopamine
- Dobutamine
- Epinephrine (adrenaline)
- Norepinephrine (noradrenaline)
- Vasopressin

The duration of action of medicine is selected to be four hours from the administration time of the medicine. Other periods are considered as not having vasopressor medication.

Ventilator periods

Ventilator is a machine that assists patient in breathing. It moves air into lungs and out of the lungs. Knowledge when patient is connected to the ventilator and disconnected from the ventilator is available in the database. Patient's ventilator period starts when the patient is connected to the ventilator and it continues until the dataset has

information that patient is disconnected from the ventilator. During other periods patient is not considered to be in the ventilator. Knowledge about ventilator treatment is used to select data periods that simulate general ward data. Mechanical ventilation is not used in the general ward.

3.1.4 Resampling

Vital sign parameters are continuously measured and recorded in the dataset at fixed time intervals. However, the sampling rates and measurement times vary between parameters. Intermittent data has also varying recording times. In order to align data in time, all parameters are resampled into the same intervals. After the resampling only one parameter value exists within a defined sampling interval. The selection of resampling interval is based on the measured sampling rates of vital sign parameters. The aim is to select sampling rate so that it is high enough to preserve most of the original information but does not create too many missing values that would require imputation. Table 4 lists sampling rates of continuously measured physiological parameters. Data is measured from the development set patient stays that were extracted from the SQL database (3318 patient stays). It can be seen from the table, that continuous parameters average measurement intervals are around 2 minutes. For all continuous parameters, 99 % of measurement time intervals are shorter than 3 minutes 37 seconds. Based on the measurements of Table 4, a sampling rate of 5 minutes was chosen for this study.

Table 4. Vital sign parameters' measurement intervals

Parameter	**Measurement count**	**Sampling rate mean**	**Sampling rate 99% percentile**
SysBP	14 125 112	1min 49s	3min 37s
MeanBP	14 071 827	1min 49s	3min 37s
DiaBP	14 076 486	1min 49s	3min 37s
HR	14 718 670	1min 55s	3min 37s
RR	9 928 852	2min 25s	3min 37s
SpO2	13 952 134	2min 01s	3min 37s
Temp	5 900 532	2min 00s	3min 21s

Table 5. Intermittent parameters' measurement intervals

Parameter	Measurement count	Sampling rate mean	Sampling rate 99% percentile
GSCe	14 113	1d 1h 17min 18s	5d 2h 35min 35s
GSCv	14 127	1d 1h 17min 19s	5d 2h 17min 13s
GSCm	14 110	1d 1h 18min 29s	5d 1h 51min 44s
Cr	16 798	23h 11min 7s	1d 2h 4min 1s
K	66 143	6h 3min 4s	1d 0h 37min 0s
Na	66 265	6h 5min 36s	1d 0h 38min 0s
WBC	19 756	20h 1min 27s	1d 1h 30min 0s
HCT	77 523	5h 17min 9s	1d 0h 24min 0s

Parameters are resampled at 5 minutes frequency using the following methods:

- Vital sign parameters (SysBP, MeanBP, DiaBP, HR, RR, SpO2, Temp) are resampled using the median value from the 5 minutes time window. If there are no measurement values available within the 5 minutes time window, the result is a missing value.
- Intermittent data (lab data, GCSe, GCSv, GCSm) and patient treatment periods (supplemental oxygen, pressurized supplemental oxygen, ventilator and vaso-pressor medication) are resampled using the last value from the 5 minutes time window. If there are no values available within the 5 minutes time window, the result is a missing value.

3.1.5 Handling missing values

After the resampling, the dataset contains missing values. A forward-fill imputation is implemented for the following intermittent parameters. In the forward-fill imputation the last available value is repeated:

- Lab data (Cr, K, Na, WBC, HCT) are forward-fill imputed at most 26 hours. It means that laboratory test is considered to be valid 26 hours. Laboratory measurements are typically done at least once a day in the hospital. From the Table 5 it can be seen that for all laboratory data measurements, 99 % of measurement time intervals are shorter than 26 hours and 4 minutes. 26 hours is selected because it matches the sampling rate 99th percentile.

- GCS subscores (GSCe, GSCv, GSCm) are forward-fill imputed without limits. I.e. the last value is treated valid until new value is available.

Vital sign parameters (SysBP, MeanBP, DiaBP, HR, RR, SpO2, Temp) are not imputed. This strategy is selected because imputation of vital sign parameters would affect not only the input features but also the prediction target. Prediction target is defined by NEWS score and NEWS is calculated from the vital sign parameters. Vital sign parameters can change fast and it would be difficult to understand the reliability of the results if target value was also affected by the imputation. GCS subscores are also used in NEWS calculation but it is mandatory to impute them because they are only occasionally recorded in the dataset. By definition (chapter 3.1.3), patient treatment periods (supplemental oxygen, pressurized supplemental oxygen, ventilation and vasopressor medication) do not have missing values.

Data periods which have missing values after the resampling and above described imputation are excluded and not utilized in this study. The advantage of this strategy is that it avoids adding artificial data values, the disadvantage is that it decreases the size of the dataset. In general, the strategy that discards data if any parameter has missing values can also create a bias because only a subset of data is used. For example, in the medical domain, the requirement that patient must be monitored by a specific parameter might select only those patients who have a certain medical condition.

The selected strategy means that this study uses data only from those patient stays that have measurements in all previously listed vital sign parameters, lab data and GCS subscores. If patient stay has any parameter which does not have measurement values at all, the patient stay is discarded. The assumption is that parameters used in this study are routinely measured for different kind of patients in the ICU and requiring presence of them does not create a bias in data. Note that if patient stay has measurements in all parameters but there are some time periods where values are missing, then only those time periods are excluded.

Table 6 shows how many patient stays have at least one measurement for parameters used in this study. These statistics has been measured from the 4419 patients stays that

were extracted from the ICU database. It can be seen from the table that all other parameters are commonly measured among patient stays except body temperature which is measured only in less than half of the patient stays. The reason is that database actually contains several different body temperature parameters and it depends on the patient case which one is used. The body temperature can be measured in multiple ways and the temperature parameter which is used in this study is the one which is measured for largest amount of patient stays. It is a central temperature which is measured by urine catheter. Even though it is measured for largest amount of patient stays, it is still measured only in 1820 patient stays which significantly decreases the size of the dataset of this study. Different temperature parameters were not merged together because different measurement methods may produce slightly different values.

Table 6. Number of patient stays that have at least one measurement in a parameter

Parameter	**Nr of patient stays that have at least one parameter measurement**
SysBP	4 338
MeanBP	4 342
DiaBP	4 337
HR	4 416
RR	4 197
SpO2	4 415
Temp	**1 820**
GSCe	4 279
GSCv	4 280
GSCm	4 277
Cr	4 375
K	4 331
Na	4 382
WBC	4 404
HCT	4 406

3.1.6 Calculating the NEWS

The dataset is not pre-labeled but labels are created as a part of the study. The National Early Warning Score (NEWS) is selected to represent the patient's medical condition. The NEWS scoring system uses the following parameters for calculating the score:

- Respiratory rate
- SpO2, oxygen saturation of arterial blood
- Body temperature
- Systolic blood pressure
- Heart rate
- Level of consciousness
- Knowledge if supplemental oxygen is given

NEWS system calculates a subscore for each parameter. The total score is a sum of subscores. When calculating subscores, NEWS system compares measured parameter values to parameters' normal value ranges. The bigger is the difference between the measured value and the normal value range, the higher is a subscore. Table 7 shows how NEWS subscores are determined from the measured parameter values. Glasgow Coma Scale values < 15 are mapped to altered mentation in NEWS which is scored as 3 by NEWS level of consciousness. The same mapping is used by Smith et al. (2013 p.4).

Table 7. NEWS calculation

Physiological parameter	**Score**						
	3	**2**	**1**	**0**	**1**	**2**	**3**
RR (1/min)	≤ 8		9-11	12-20		21-24	≥ 25
SpO2 (%)	≤ 91	92 – 93	94-95	≥ 96			
Supplemental oxygen		Yes		No			
SysBP (mmHg)	≤ 90	91–100	101-110	111-219			≥ 220
HR (1/min)	≤ 40		41-50	51-90	91-110	111-130	≥ 131
Consciousness				Alert GCS = 15			CVPU GCS ≤ 14
Temp (°C)	≤ 35.0		35.1-36.0	36.1-38.0	38.1-39.0	≥ 39.1	

NEWS subscores vary in a range 0 - 3. The range of the total NEWS score is 0 – 20. The higher is the score, the sicker is the patient. 0 represents a normal medical condition. A NEWS value 7 is commonly used as a threshold at hospitals to decide that patient needs to be transferred to ICU or other higher-dependency care unit (Royal College of Physicians 2017).

In this study, the NEWS is first calculated for each 5 minutes time interval using the resampled parameter values. After that, 15 minutes median of NEWS is calculated using a moving window. 15 minutes median value which is stored at time t, is a median of NEWS 5 minutes values from times t, t-5min and t-10min. The purpose of the median calculation is to reduce the effect of artifacts. For example, SpO2 value can change significantly if patient moves which may cause spikes in the NEWS data. Short-term spikes are filtered by median calculation. This study uses only NEWS 15 minutes median values, NEWS 5 minutes resolution values are not used.

3.1.7 Prediction task

The aim of the study is to predict patient deterioration using patient's medical information from the past. The target environment is a simulated general ward which supports continuous monitoring of vital sign parameters. Patient deterioration is determined by the NEWS score. Deterioration can happen from any clinical reason.

The following time windows are used in the prediction: history window from which input features are calculated, prediction window from which the prediction target is determined and a gap between the history window and the prediction window. The purpose of the gap is to provide enough time for the clinical intervention. The size of the history window should be large enough to capture the trends in patient condition. On the other hand, the history window should not be too large, because shorter history window makes it possible to start the prediction earlier. The gap should be large enough to give time for the clinical intervention. On the other hand, the larger is the gap, the harder is the prediction task. This thesis uses history window of 1 hour, gap of 1 hour and prediction window of 2 hours. The prediction is done at 15 minutes interval if patient data is

available and patient is not yet deteriorated. Figure 1 displays time windows that are used in the prediction. The prediction is done at time t.

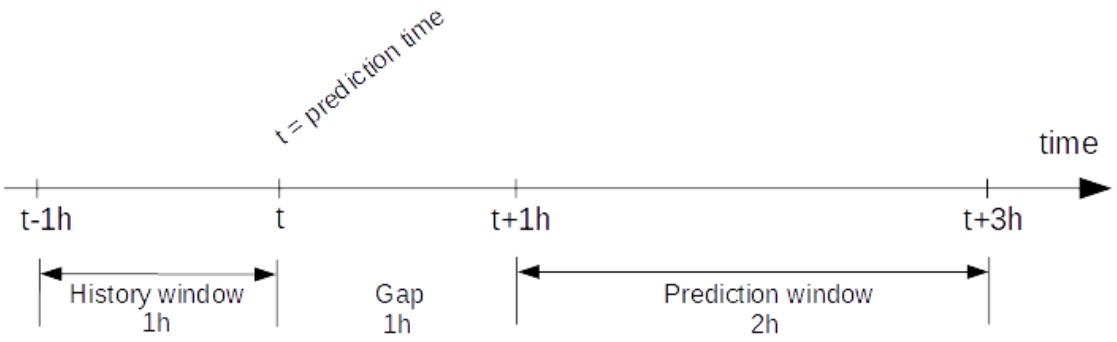

Figure 1. Prediction time windows

The target label is determined from the prediction window. The patient is deteriorated if the maximum NEWS in the prediction window is greater than or equal to 7. If the maximum NEWS is less than 7, patient is not deteriorated. This definition means that the prediction task is a binary classification problem. Used threshold is adopted from the NEWS documentation which recommends that score value of 7 should be used as a threshold for high level clinical-alert which usually triggers patient transfer to higher dependency care area (ICU) (Royal College of Physicians 2017).

3.1.8 Features

Table 8 lists statistical features that are calculated from the parameter data and used as model features. The described features are calculated for all listed parameters on the table. For example, 8 statistical features are calculated for the SysBP. All features are calculated from the history window using 5 minutes resampled values except gender and age which are static data and NEWS which is 15 minutes median. The purpose of the statistical features is to summarize the physiologial parameters' behavior in the history window and to decrease the number of model's features. The total number of features is 75 which consists of 73 statistical features plus age and gender.

Table 8. Statistical features

Input parameter	Calculated features (from the history window)
SysBP, MeanBP, DiaBP, HR, RR, SpO2, Temp, NEWS	First (oldest) value Last (newest) value Difference between last and first value (last - first) Minimum value Maximum value Average value Median value Standard deviation
GSCe, GSCv, GSCm, Cr, K, Na, WBC, HCT, Is supplemental oxygen given	Last (newest) value

3.1.9 Finding data periods that simulate general ward

The aim of the thesis is to predict patient deterioration in the hospital's general ward which supports continuous monitoring of vital sign parameters. Continuous measurement data from the general ward is not yet available and that's why ICU data is used to simulate the general ward. The benefit of ICU data is that it provides continuous measurements of several physiological parameters. The disadvantage is that patients in the ICU are typically in a different medical condition than patients in the general ward. In addition, patient treatment in ICU (used medication and devices) differs from the general ward. These differences mean that patient's physiological parameters don't necessarily behave in a same way in the ICU and in the general ward. Because of that, only a subset of data is selected from the ICU dataset and used in this thesis to simulate the general ward.

Features and prediction targets are calculated from the time windows described by the chapter 3.1.7. This chapter describes which patient data periods are valid for positioning these time windows. Features and prediction targets are calculated only from those data periods that simulate general ward data. Data period is used if all of the following conditions are fulfilled. Conditions 1-3 implement the simulation of ward data, condition 4 is not related to the simulation but to the selected imputation strategy (see chapter 3.1.5):

1. Data is not from the first 24 hours of ICU stay. Patients are admitted to ICU when they have severe illness or injury. First 24 hours is excluded, because patients' medical condition does not correspond to ward patients' typical medical condition.
2. Patient is not deteriorated within the history window or gap. Patient is deteriorated if maximum NEWS is greater than or equal to 7. The task of this thesis is to predict if non-deteriorated patient is going to deteriorate. After the deterioration, patient gets intensive treatment which affects the behavior of physiological parameters. This situation does not simulate patient in the general ward.
3. Patient does not have any of the following treatments within a history window or gap: mechanical ventilation, vasopressor medication or pressurized supplemental oxygen therapy. These treatments affect the behavior of physiological parameters and they are not used in the general ward. Note that these treatments are allowed in the prediction window.
4. Input parameters (vital signs, lab data, GSCe, GSCv, GSCm) do not have any missing values within the history window, gap or prediction window after the imputation described by chapter 3.1.5. Data is only partially imputed and all periods which still have missing values after the imputation are discarded.

Data samples are searched using the following logic: first history window, gap and prediction window are positioned to the beginning of the second day of patient stay data. After that the above listed conditions are checked. If all conditions are fulfilled, feature vector and target label are calculated from the current window positions. If conditions are not fulfilled, nothing is done. After that, history window, gap and prediction window are forwarded 15 minutes in time, conditions are rechecked and if conditions are fulfilled, new feature vector and target label is calculated. This process is repeated until the end of patient data is reached. After that the next patient is processed similarly.

The above described process produces varying number of samples per patient. From some patients it is not possible to find any samples, from other patients it is possible to find several samples. The number of samples per patient is not limited. The number of positive or negative samples per patient is not limited either which means that it is possible to get several positive samples from the same patient. The data which is produced

by the above described process constitutes final dataset of this thesis. The produced dataset is used to train and test machine learning models. The chapter 3.1.10 displays statistics of the created dataset.

3.1.10 Dataset statistics

This chapter presents statistics about the created dataset. The total size of the dataset that simulates general ward is 64 084 samples. Samples are gathered from 917 patient stays. Table 9 and Table 10 summarize statistics of the dataset.

Table 9. Statistics about the dataset which simulates general ward

	Development set	**Test set**
Size (nr of samples)	47 084	17 000
Size (percentage)	73 %	27 %
Nr of patient stays	679	238
Samples per patient stay average	69,3	71,4
Samples per patient stay median	43	41
Number of positive samples	6 014	1 829
Positive samples (percentage)	12,8 %	10,8 %
Nr of patient stays that have positive samples	397	140
Nr of patient stays that have negative samples	641	219

Table 10. Patient demographics in the dataset which simulates general ward

	Development set	**Test set**
Patient age average	57,2	56,2
Patient age median	60	60
Male patients	459 (67,6 %)	152 (63,9 %)
Female patients	220 (32,4 %)	86 (36,1 %)

Figure 2 displays the distribution of max NEWS in the prediction window for the development set samples. If max NEWS in prediction window is equal or greater than 7, a sample is positive. It can be seen from the figure that 5 is the most common value and most of the samples have score between 3 and 6. Most of the deteriorating patients have a score of 7 which equals to the used threshold. It may be difficult for models to separate patients that have a score 7 from patients that have e.g. a score of 6.

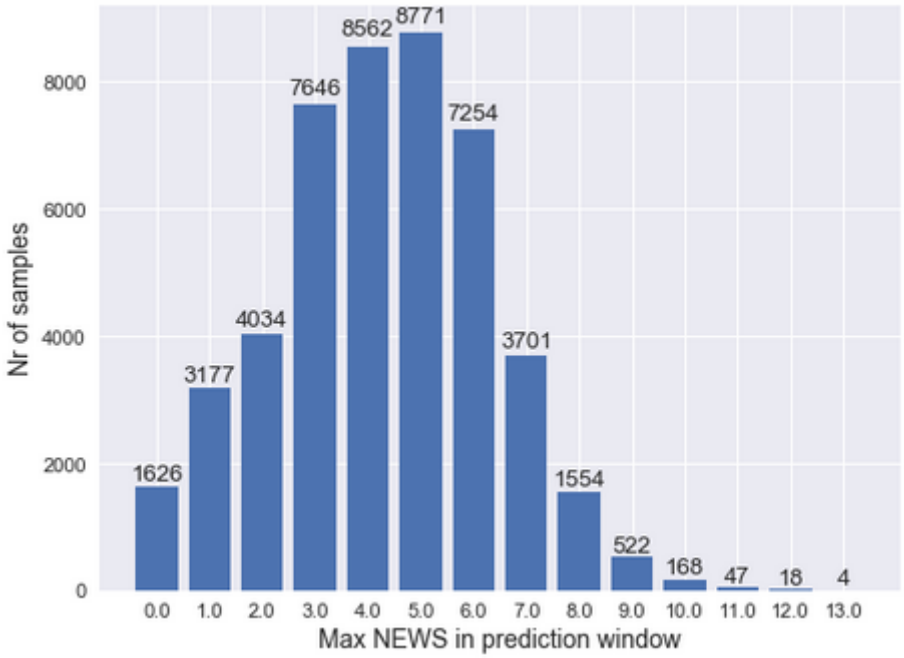

Figure 2. Distribution of max NEWS in the prediction window for development set samples

Figure 3 displays the distribution of max NEWS in prediction window for the test set samples. In the test set, 4 is the most common value. Like in the development set, most of the samples have score between 3 and 6.

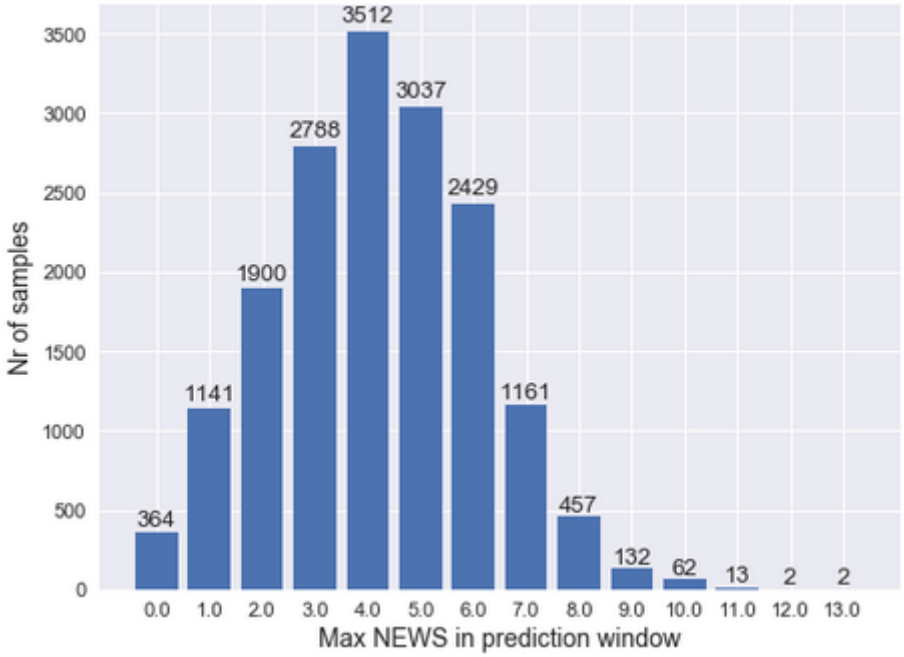

Figure 3. Distribution of max NEWS in prediction window for test set samples

Figure 4 is a frequency chart which displays how much positive samples' NEWS value changes between the history window and the prediction window. X-axis displays the difference between the maximum NEWS in the prediction window and the last NEWS in the history window. Y-axis displays how many samples have the change. For positive samples, the score can only increase. It can be seen from the chart that for most of the deteriorating patients NEWS value increases 3 or more points. This information supports the idea that it may be possible to discriminate deteriorating patients from non-deteriorating patients. If the change in NEWS for positive samples had been mostly 1 or 2 points it would have probably been more difficult to separate positive and negative samples from each other.

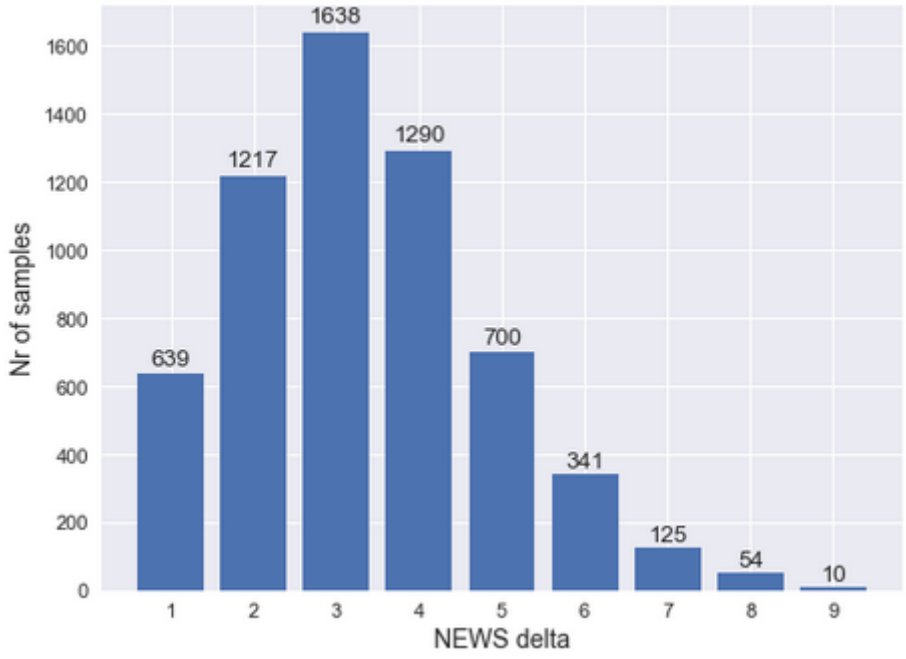

Figure 4. NEWS delta for development set positive samples (NEWS delta = max NEWS in prediction window – last NEWS in history window)

Figure 5 displays difference between the maximum NEWS in the prediction window and the last NEWS in the history window for negative samples in the development set. For negative samples, the score can either increase or decrease. For most of the samples, the score remains the same or slightly increases. It is worth noting that there are some patients whose NEWS increases a lot, even 6 points. These samples are still negative in

this study even though patient's clinical situation seems to be worsening fast. Samples are negative because the max NEWS in the prediction window remains below 7. Data in Figure 4 and Figure 5 is measured from the development set.

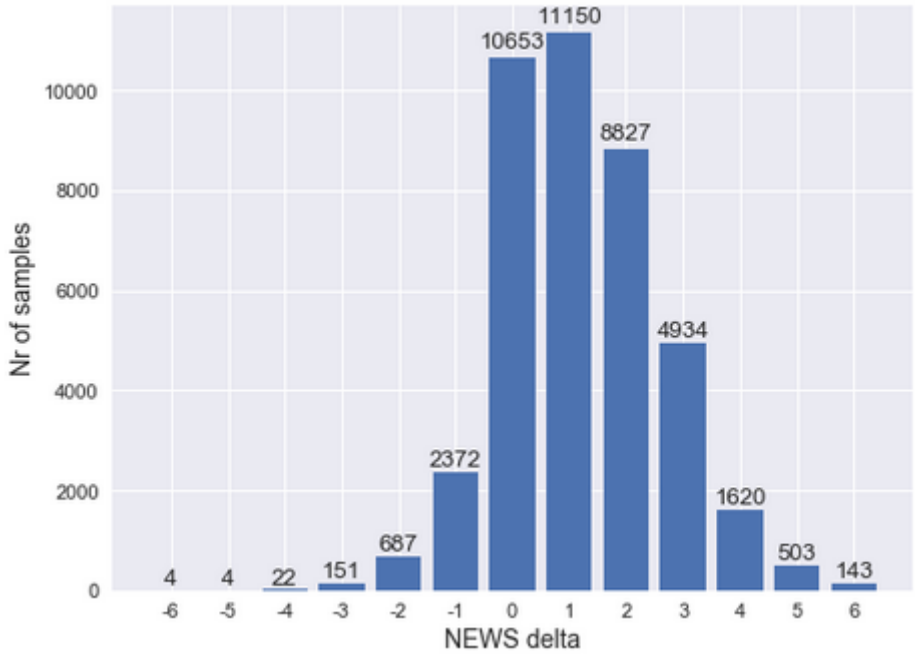

Figure 5. NEWS delta for development set negative samples (NEWS delta = max NEWS in prediction window – last NEWS in history window)

3.2 Machine learning methodology

This chapter describes the machine learning modeling methods that have been used in this study. The chapter explains methods for training the models, searching the best model and evaluating the final performance of the best model. The chapter also describes the baseline and performance metrics that are used in this study. Finally, the internal functionality of tested models are explained.

3.2.1 Development set and test set

Dataset is randomly split into development set and test set in the original SQL database so that development set contains 75% of patient stays and test set contains 25% of patient stays. However, because this study uses only a subset of the original ICU data,

the split of the final dataset is slightly different. The final development set contains 73 % of samples and the final test set contains 27 % of samples as described by the Table 9. The size of the development set is 47 084 samples, the size of the test set is 17 000 samples. Development set samples have been gathered from 679 patient stays, test set samples have been gathered from 238 patient stays. It is important to notice that all samples of one patient stay are either in the development set or in the test set. The development set is used for model training and for selecting the best model. The test set is used only for final performance evaluation of the best model. All tested models used the same split for development and test set.

3.2.2 Grid search with cross-validation

The best parameters for tested models are searched using a grid search with 10-fold cross-validation in the development set. Cross-validation folds are selected so that all samples of one patient stay are either in the training fold or test fold. If samples from the same patient were used both for training and testing the model, evaluated generalization performance could be too optimistic (Müller & Guido 2016 p. 261). Cross-validation folds were equal for all tested models. Grid search is done over pre-selected parameter values that are described by the chapter 5.1. Only the development set is used for selecting the parameters and finding the best model, the test set is used only for final evaluation of the best model.

3.2.3 Performance metrics

Plenty of evaluation metrics are available for the classification. The selection of the best metric depends on the application, the high-level goal and how imbalanced the dataset is. This chapter describes the metrics that are reported by this thesis and used to evaluate the model performance.

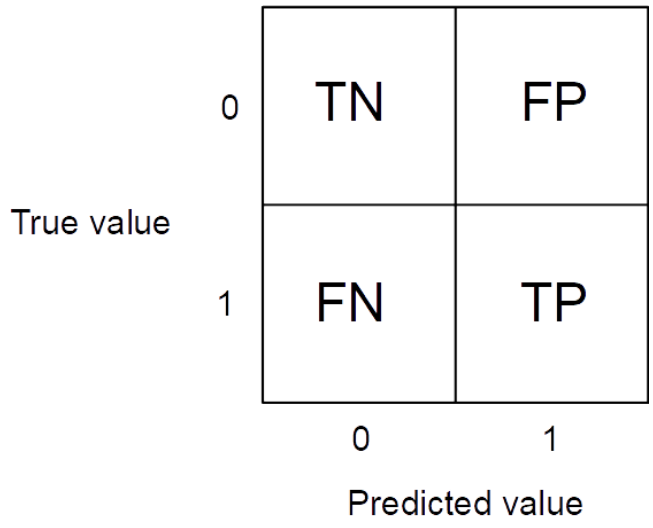

Figure 6. Confusion matrix

Figure 6 presents a confusion matrix where classification metrics are derived from. Definitions:

TN True negative. Negative class correctly classified as negative.

TP True positive. Positive class correctly classified as positive.

FN False negative. Positive class incorrectly classified as negative.

FP False positive. Negative class incorrectly classified as positive.

The following metrics are derived from the confusion matrix and used by this study:

Precision, positive predictive value (PPV)	$\frac{TP}{TP+FP}$
Sensitivity, Recall, True positive rate (TPR)	$\frac{TP}{TP+FN} = 1 - FNR$
Specificity, True negative rate (TNR)	$\frac{TN}{TN+FP} = 1 - FPR$
False positive rate (FPR)	$\frac{FP}{FP+TN} = 1 - TNR$
False negative rate (FNR)	$\frac{FN}{FN+TP} = 1 - TPR$
F1-score	$2x\frac{PPV\ x\ TPR}{PPV + TPR}$
Positive likelihood ratio (LR+)	$\frac{TPR}{FPR}$
Negative likelihood ratio (LR-)	$\frac{FNR}{TNR}$

Classification models use a decision threshold for making decisions between positive and negative classes. The above listed metrics report the performance for a fixed threshold. In addition to them, receiver operating characteristics curve and precision-recall curve can be used to display the model performance for all possible thresholds. The precision-recall curve displays all possible trade-offs between the precision and the recall (sensitivity, TPR), while receiver operating characteristics curve displays all possible trade-offs between the recall and the FPR (1 - specificity). The information of these curves can be summarized by calculating the area under the curve. Area under the receiver operating characteristics curve (AUROC) and area under the precision-recall curve (AUPRC) are reported by this study. AUROC and AUPRC vary in a range from 0 to 1, where 1 is the best possible value. Actual curves are plotted in the final test set performance evaluation. In the development set cross-validation, areas under the curves (AUROC, AUPRC) are reported.

Metrics that were most often reported in the literature review studies were sensitivity, specificity and AUROC. Sensitivity measures a probability that positive class is correctly classified as positive, specificity measures a probability that negative class is correctly classified as negative. In the medical domain, it is important to identify as many deteriorating patients as possible. I.e. it is important to have high sensitivity. Both sensitivity and specificity are reported by this study. A shortcoming in the literature review studies was that precision was not always reported. Precision measures the probability that a sample classified as positive is actually positive. Precision is used in a situation when it is important to avoid false positives which is often the goal in the medical domain (Müller & Guido 2016 p. 285). Low precision means many false positives (i.e. many false alarms) which is a known problem in the medical domain. Having too many false positives (false alarms) may mean that predictions are in practice ignored by clinical staff. Precision is an interesting metric also because high sensitivity and high specificity do not mean that the precision would be high. The disadvantage of precision and also AUPRC is that they cannot be easily compared between different studies because they are affected by the prevalence of the dataset. If the prevalence of the study is increased, precision and AUPRC are likely increased too (Positive and negative predictive values, 2019). Precision and AUPRC for the model which makes random predictions is the proportion of positive samples in the dataset. In the medical domain, datasets are often

imbalanced which means that precision and AUPRC values tend to be low. Both precision and AUPRC are reported in this study. In addition to precision-recall curve and AUPRC, recall and precision can also be summarized by using f1 score which is a harmonic mean of recall and precision (Müller & Guido 2016 p. 285).

AUROC was reported by all investigated literature review studies. The benefit of sensitivity, specificity and AUROC is that they do not depend on the prevalence of the dataset. Random predictions always results in AUROC of 0,5 regardless of how imbalanced the dataset is. AUROC is highly recommended metric for imbalanced datasets (Müller & Guido 2016 p. 298). According to Müller & Guido (2016 p. 297), AUROC can be interpreted so that it equals to the probability that randomly selected positive sample will be ranked higher than the randomly selected negative sample. In general, AUROC values greater than 0.8 are considered as indicating good discrimination (Smith et al 2013 p. 5) or excellent discrimination (Mandrekar 2010 p. 1316). AUROC range from 0,7 to 0,8 is described to indicate reasonable (Smith et al 2013 p. 5) or acceptable discrimination (Mandrekar 2010 p. 1316). In this thesis, AUROC is reported in the cross-validation and test set results and it is also used as a criteria for selecting the best model and best model parameters in a grid search cross-validation.

Two additional metrics that are reported by this study are positive and negative likelyhood ratios which are widely used in medical diagnostics. Positive likelyhood ratio (LR+) is a probability that positive class is classified as positive divided by probability that negative class is classified as positive. I.e. how much more likely it is to have positive classification for the positive class than for the negative class. Bigger LR+ value is better. LR+ should be greater than 1. Negative likelyhood ratio (LR-) is a probability that positive class is classified as negative divided by probability that negative class is classified as negative. I.e. how much more likely it is to have negative classification for the positive class than for the negative class. Smaller LR- values are better. LR- should be less than 1. LR+ and LR- are calculated using sensitivity and specificity. Likelyhood ratios have an advantage that they do not depend on the prevalence of the dataset (Positive and negative predictive values, 2019).

3.2.4 Baseline

A random prediction can be thought as a first baseline. Random prediction provides AUROC of 0,5 and AUPRC which equals to the proportion of positive samples in the dataset (in this study, 12,8% in the development set and 10,8% in the test set). A developed machine learning model should be much better than a random prediction.

A random prediction as a baseline is not necessarily very interesting. More interesting question is that what is the performance of the developed machine learning model compared to the current clinical practices. I.e. can the developed model provide any benefit in real life. A random prediction as a baseline can not answer to this question because patient treatment decisions are not done randomly at the hospital. Finding a good baseline that has practical relevance is often challenging.

The main baseline of this study is a simple rule which utilizes NEWS scoring system's medium level alert. According to recommendations of Royal College of Physicians (2017), NEWS value of 5 should be used as a threshold for medium level clinical alert which is an indication of potential serious clinical deterioration. The response to medium level alert is urgent clinical review. In their final report, Royal College of Physicians refer to study (Smith et al. 2013) which analyzed NEWS system's ability to identify patients who were at risk in clinical deterioration. In their reference, Royal College of Physicians report that "it is clear that a NEW score of 5 or more is associated with increased risk of a serious clinical outcome". The baseline of this study is a simple rule which uses this threshold and predicts that patient will deteriorate if the last NEWS in the history window is greater than or equal to 5. NEWS medium level alert has much better performance than a random prediction and provides practically more relevant baseline. A machine learning model should be better than this baseline in order to provide benefit in practise.

3.2.5 Handling dataset imbalance

In this study, dataset imbalance is compensated by calculating class weights which are used by the loss function to penalize class errors differently during the training. Class

weights are calculated using the following equation which provides weights that are inversely proportional to class frequencies (RandomForestClassifier, 2017):

$$class\ weight = \frac{count(samples)}{count(classes)\ x\ count(class\ occurrences)} \quad (1)$$

where

- count(samples) is the size of the dataset
- count(classes) is number of different classes in the dataset
- count(class occurrences) is number of occurrences of a class whose weight is calculated

A minority class will have a higher class weight which means that misclassified minority class will be penalized more than misclassified majority class.

Class weights are in practise implemented using scikit-learn tools as follows:

- if classifier supports class_weight parameter, it is set to 'balanced'.
- if classifier doesn't support class_weight parameter, a compute_class_weight() method with class_weight='balanced' argument is used to calculate sample weights that are given to model's fit()-method.

3.2.6 Performance evaluation using the test set

The best model is searched using grid search with 10-fold cross-validation in the development set as described by chapter 3.2.2. After that, the final performance of the best model is evaluated in the test set. The model is trained on the whole development set and the performance is evaluated on the test set. Test set is not otherwise used.

3.2.7 Tested models

The main modeling interest of this thesis was in decision tree based ensembles; random forest and gradient boosted regression trees. They are known to have a good performance both in classification and regression. Their advantages are that they don't need scaling of data, they can be used with linear and nonlinear data, they work well with both numerical and categorical features and they provide feature importance

information. Two other models, that were included in this study, were logistic regression and kernelized support vector classifier. Logistic regression is more simple model which uses linear function of features for separating classes while support vector machine allows more complex nonlinear decision boundaries. The following chapters give a brief introduction to principles of operation of tested models.

3.2.8 Decision tree

Decision tree is a structure which consists of series of if/else questions which lead to an answer. The goal of the decision tree is to find the correct classification or regression answer using as few questions as possible. In the classification the correct answer is the class label. The question in the tree is called a node, the answer is called a terminal node or a leaf. Each node in the tree splits the data into two parts with a question that uses one of the features and a threshold. Series of questions in a tree result in recursive partitioning of data.

In the decision tree training, the aim is to construct recursive questions that lead to the correct answer as fast as possible. In the training process, the algorithm searches for a question that best splits the training data. In the binary classification, the aim is to find a question that best separates positive samples from the negative samples. The quality of a split can be measured using different metrics. In scikit-learn the options are Gini impurity and Entropy. Gini impurity, which is used in this study, is calculated for a set of items using the following equation: (Decision tree learning, 2019), (Decision Trees, 2009)

$$G = 1 - \sum_{i=1}^{J} {p_i}^2 \qquad (2)$$

where

- J is the number of classes
- p_i is fraction of items labeled with i in the set

The training algorithm creates candidate splits for different features and threshold values. Gini impurity is then calculated for both subsets of a candidate split and calculated values are combined to give the overall quality of the split. The total Gini impurity of a

split is calculated using the following equation. The best split is the one which minimizes the total Gini impurity. (Decision Trees, 2009)

$$G_{total} = \mathrm{G}_{\mathrm{left}} \frac{\mathrm{n}_{\mathrm{left}}}{N} + \mathrm{G}_{\mathrm{right}} \frac{\mathrm{n}_{\mathrm{right}}}{N} \quad (3)$$

where

- $\mathrm{G}_{\mathrm{left}}$ is Gini impurity of a left split
- $\mathrm{G}_{\mathrm{right}}$ is Gini impurity of a right split
- $\mathrm{n}_{\mathrm{left}}$ is number of samples in a left split
- $\mathrm{n}_{\mathrm{right}}$ is number of samples in a right split
- N is total number of samples in both splits ($\mathrm{n}_{\mathrm{left}} + \mathrm{n}_{\mathrm{right}}$)

The result of the training process is a binary decision tree. If not restricted, the process is repeated until pure terminal nodes are found. Pure terminal node contains only training samples that have the same target value. Constructing a decision tree until all terminal nodes are pure typically creates a deep tree which is complex and likely overfits to the training data. In practice it is necessary to restrict the complexity of a decision tree. In scikit-learn, the complexity of a tree is controlled by setting parameters that define when the tree construction is stopped. Available options include setting a maximum depth of a tree (max_depth), setting minimum number of samples required in a terminal node (min_samples_leaf) or setting the maximum number of terminal nodes (max_leaf_nodes). With imbalanced datasets, it is recommended to use dataset balancing in order to avoid biased model. (Decision Trees, 2009), (Müller & Guido 2016 p. 76)

Predicting a class label with a decision tree means traversing through tree's questions until terminal node is reached. The predicted label is a majority label of terminal node's training samples or in case of pure terminal node, the only label that is left (Müller & Guido 2016 p. 76). Alternatively, a probability of each class can be predicted by calculating the fraction of samples in the terminal node that have the same class (Decision Trees, 2009). It means that, in the pure terminal node, the model is 100% sure about its prediction.

3.2.9 Random forest

Random forest is a collection of decision trees. Random forest tries to avoid overfitting tendency of a decision tree by building several decision trees that are slightly different and by averaging their results. The idea behind the random forest is that although individual decision trees may overfit to the training data, the overall overfitting is reduced by averaging results of all decision trees. (See Müller & Guido 2016 p. 85)

There are two sources of randomness when building the random forest. First, each decision tree in a forest is built using a different dataset. Training datasets for individual decision trees are created by taking a boostrap sample of the original training data (random sample with replacement). Created boostrap sample is as big as the original dataset but some of the original data points are repeated in it and some data points are missing. Another source of randomness is that when searching for the best split of a node, only a random subset of features are considered. Random subset of features is selected separately for each node of a decision tree. The number of considered features can be set in scikit-learn by setting the max_features parameter. In random forest, each decision tree is an independent model which makes it possible to do training and make predictions in parallel. (Ensemble methods, 2012)

The main parameters for a random forest include number of decision trees and number of features that are considered in each split. Larger number of decision trees is always better but require more computation time and memory. After a certain point, adding new trees doesn't significantly improve the results anymore. Maximum number of features considered in each split controls how random decision trees in a forest are. Also the parameters that control the complexity of an individual decision tree, like already mentioned max_depth, min_samples_leaf and max_leaf_nodes, are available in scikit-learn for a random forest. In scikit-learn, the prediction of a random forest is created by averaging probabilistic predictions of individual decision trees. (Ensemble methods, 2012)

3.2.10 Gradient boosting classifier

Gradient boosted regression trees is also an ensemble of decision trees. It can be used both for classification and regression. The idea is to build a collection of simple decision trees in a sequential way so that each added decision tree tries to correct the mistakes of the previously built ensemble. The difference between the random forest and the gradient boosted regression trees is that in the random forest decision trees are independent models that are trained using a random sample of data and random subset of features in each node. In the gradient boosting, models are not created randomly but sequentially. By default gradient boosting does not use randomization although in scikit-learn maximum number of features considered in each split can be defined by max_features parameter and the fraction of samples used to train individual decision trees can be defined by subsample parameter. In gradient boosting, decision trees are usually shallow to avoid overfitting. Because of a sequential structure of the model the training process can not be parallelized and may be time consuming.

The term boosting refers to machine learning techniques which combine weak learners, like simple decision trees, into a strong one. Gradient boosted regression trees generalizes boosting to arbitrary differentiable loss functions (Ensemble methods, 2012).

Gradient boosting classifier is built by adding new decision trees to the ensemble one at a time. When new tree is added to the model old trees are left unchanged. The goal of the process is to add new trees to the ensemble so that each tree improves the performance of the ensemble. Usually the process is stopped when predefined number of trees have been added. The ensemble model can be expressed using the following equation (Ensemble methods, 2012):

$$\hat{y} = F(x) = \sum_{m=1}^{M} \gamma_m f_m(x) \qquad (4)$$

where

- $F(x)$ is ensemble model
- x is matrix of feature vectors
- $f_m(x)$ is a decision tree (weak learner)

- M is number of decision trees in the ensemble
- $\hat{y}$ is vector of predicted targets
- γ_m is a step length multiplier

When new decision tree is added to the ensembe, the aim is to create a better ensemble $F(x)$. The performance of the ensemble is measured by a loss function L. The total loss is the average of losses of individual observations (Parr & Howard):

$$L(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N} L(y_{i,} F(x_i)) \quad (5)$$

where

- y is a vector of correct targets
- $\hat{y}$ is vector of predicted targets
- x_i is a feature vector of a single observation
- y_i is a correct target for a single observation
- N is the number of observations

In the training process, the overall loss is minimized using an optimization process called gradient descent. The aim of the process is to gradually reduce the loss when new trees are added to the ensemble. The loss function can be any differentiable function. When a new weak learner is added to the ensemble, it is trained using a negative gradient of a loss function as a label data. I.e. adding a new weak learner to the ensemble means adding weak learner's approximation of loss function's negative gradient to the output of the ensemble model. Because negative gradient of a loss function points to the direction of minimized loss function the process reduces the overall loss of the ensemble step by step. (Parr & Howard)

New ensemble model at step m is:

$$F_m(x) = F_{m-1}(x) + \gamma_m f_m(x) \quad (6)$$

where newly added weak learner (decision tree) approximates loss function's negative gradient (Ensemble methods, 2012):

$$F_m(x) = F_{m-1}(x) - \gamma_m \sum_{i=1}^{N} \nabla_F L(y_{i,} F_{m-1}(x_i)) \quad (7)$$

Step length multiplier γ_m is calculated by the algorithm to optimize the effect of the step, e.g. to avoid taking too big steps that would pass the minimum of a loss function.

Number of estimators can be used as a regularization parameter. It is specified by a parameter n_estimators in scikit-learn. Unlike with random forest where larger number of decision trees is always better, in gradient boosting more estimators provide more complex model which is more likely to overfit. Another parameter which can be used to control the complexity of the ensemble is a learning rate which is also called a shrinkage (Friedman 2001 p. 1203). It restricts the effect of decision trees. It is a constant value which is the same for all trees:

$$\hat{y} = F_m(x) = F_{m-1}(x) + \eta\gamma_m f_m(x) \quad (8)$$

where η is the learning rate which controls the contribution of weak lerners. $0 < \eta <= 1$

Number of estimator and learning rate are interconnected and grid search is used in this study to find their optimal values.

3.2.11 Logistic regression

Logistic regression is a linear classification model which estimates probabilities that a sample belongs to one of the possible output classes. Logistic regression can be used for binary and multiclass classification. This explanation focuses only on the binary classification. In linear model, the decision boundary which separates positive classes from negative classes, is a linear function of input features. In the logistic regression, a linear relationship between the input features and log odds of a positive class is assumed (Logistic regression, 2019):

$$\mathrm{l} = \log_{\mathrm{b}} \frac{p}{1-p} = w^T x + b \quad (9)$$

where

- l is log odds of a positive class
- p is the probability of a sample belonging to the positive class
- b is base of the logarithm
- $w^T x + b$ is a linear equation where x is a vector of features, w is a vector of weights and b is a bias term

In most applications b is selected to be e (Logistic regression, 2019) after which p can be resolved from the equation (9):

$$\mathrm{p(y = 1|x; w, b)} = \frac{1}{1 + e^{-(w^T x + b)}} = F(x) \tag{10}$$

where F(x) represents the logistic regression function which provides the probability of input data belonging to a positive class. Probability of a negative class is then:

$$\mathrm{p(y = 0|x; w, b)} = 1 - \mathrm{p(y = 1|x; w, b)} = 1 - F(x) \tag{11}$$

Because y is either 1 or 0, the following applies:

$$p(y|x; w, b) = F(x)^y (1 - F(x))^{(1-y)} \tag{12}$$

The equation (10) is a standard logistic (sigmoid) function which has s-shape and output values limited between 0 and 1. The output probability can be converted to a binary classification by selecting a threshold and predicting positive class if probability is above the threshold.

The weights w and bias b of linear equation are solved in model training. Typically they are calculated using maximum likelihood estimation (Logistic regression, 2019). The aim is to find w and b that maximize the following likelihood function:

$$L(\mathrm{w, b|x}) = \prod_{i=1}^{N} p(y_i|x_i; w, b) \tag{13}$$

where

- x_i is a feature vector of a single observation
- y_i is a correct target for a single observation
- N is the number of observations.

Typically log-likelihood is maximized (Logistic regression, 2019):

$$\ln L(\mathrm{w, b|x}) = \sum_{i=1}^{N} \ln p(y_i|x_i; w, b) \tag{14}$$

In the machine learning, the convention is have a loss function which is minimized. Maximizing the log-likelihood (*14*) equals to minimizing the following negative log-likelihood:

$$-\ln L(\mathrm{w},\mathrm{b}|\mathrm{x}) = -\sum_{i=1}^{N} \ln p(y_i|x_i;w,b)$$
$$= -\sum_{i=1}^{N} (y_i \ln(F(x_i)) + (1-y_i)\ln\big(1-F(x_i)\big)) \quad (15)$$

where $F(x_i)$ is predicted probability of a positive class for a single observation.

The log-loss per sample from the equation (15) is (Model evaluation: quantifying the quality of predictions, 2010):

$$L(y_i,\hat{y}_i) = -(y_i \ln\big(F(x_i)\big) + (1-y_i)\ln\big(1-F(x_i)\big)) \quad (16)$$

The total loss is the average of losses of individual observations (*5*). Weights w and bias b are solved by minimizing the loss over all training samples. However, no closed-form solution exists for this minimization problem. Instead, optimization techniques like gradient descent must be used.

In scikit-learn, L1 or L2 regularization can be optionally added to the loss function. Parameter C defines the strength of the regularization. In this study, best values for penalty term and parameter C are searched in grid search.

3.2.12 Support vector classifier

Support vector machines are supervised machine learning models for classification and regression. For the classification, both linear and nonlinear versions exist. Linear classifiers use linear function of input features for separating classes while nonlinear classifiers allow more complex decision boundaries. The model that is used in this study is a nonlinear kernelized classifier (SVC in scikit-learn). Kernelized support vector machines are an extension of linear support vector machines. They provide configurable kernel functions which allow them to learn nonlinear decision boundaries.

The purpose of the kernel function is to transform input data into another form. Non-linear kernels allow mapping of input features into higher dimensional space where it may be possible to separate classes using a hyperplane. The higher dimensional representation of input data allows modeling of nonlinear relationships between input

features and class labels. Two commonly used nonlinear kernels are polynomial and radial basis function (rbf) kernels. Rbf is the default kernel in scikit-learn and used in this study.

The kernel function represents a similarity between two datapoints in a higher dimensional feature space. The benefit of the kernel function is that it returns the similarity in a higher-dimensional space without a need to transform datapoints into higher-dimensional space. Transforming datapoints into higher-dimensional space and calculating similarities there would be computationally more expensive. Another benefit of kernel based algorithm is that the kernel function is configurable. In scikit-learn, it is specified by model's kernel parameter. Different kernels provide different kinds of mappings to higher-dimensional feature space. Rbf kernel function is defined by the following equation (Müller & Guido 2016 p. 100):

$$k_{rbf}(x_1, x_2) = e^{(-\gamma||x_1 - x_2||^2)} \quad (17)$$

where

- x_1 and x_2 are data points (feature vectors)
- γ is a kernel coefficient (model's gamma parameter)

Support vector machine's training process finds the decision boundary which best separates classes and maximizes the margin around the decision boundary. The margin is the distance between the decision boundary and the nearest data point. The training process learns how important each data point is in defining the decision boundary. The most important data points are on the border of the classes and they are called support vectors. Because the decision boundary is defined by support vectors which are a subset of data points, adding a new data point which is not close to the decision boundary and on the correct side, does not affect model's fit to the data (VanderPlas 2017 p.410). One of the benefits of support vector machines is the insensitivity to these outliers. In scikit-learn, support vectors are stored in model's support_vectors_ attribute and support vector importances are stored by dual_coef_ attribute (Müller & Guido 2016 p. 100).

Model's regularization parameter C specifies the trade-off between misclassification of training samples and smoothness of the decision boundary. A low C tolerates more misclassifications and makes the decision boundary smooth while high C aims at classifying all training datapoints correctly (Support Vector Machines, 2017). Another important parameter is rbf kernel's gamma. It defines how far the influence of a single training datapoint reaches (Müller & Guido 2016 p. 101). Higher gamma and higher C mean more complex model. In this study, best values for C and gamma are searched in grid search.

When predicting a class for a new data point, data point's similarity to support vectors is calculated. The classification decision depends on the similarities and importances of the support vectors.

Support vector machines are powerful models which work well on different kind of datasets. They can be used both with high- and low-dimensional data. They offer configurable kernel functions which allow them to learn complex decision boundaries. The disadvantages are that the model training can be slow with big datasets and they require careful tuning of parameters (C and gamma). They also require that all features are scaled to vary on a same range.

4 EXPERIMENT SETUPS

4.1 Tools and environment

The following development tools were used in this study:

- Jupyter interactive Python programming environment
- Numpy, Pandas, SciPy data science libraries for Python
- Matplotlib, Seaborn data visualization libraries for Python
- PostgreSQL relational database
- Scikit-learn machine learning library for the Python
- Papermill Python library for configurating and executing Jupyter notebooks
- PyCharm IDE, Pylint static code analyzer

All machine learning models were implemented using the scikit-learn library. Computing was done in two Windows 10 machines:

- Dell Laptop, i7 2,7GHz, 16 GB RAM, 512 GB SDD
- HP Omen Desktop, i7 3,7 GHz, 32 GB RAM, 512 GB SDD

4.2 Overview of implemented SW

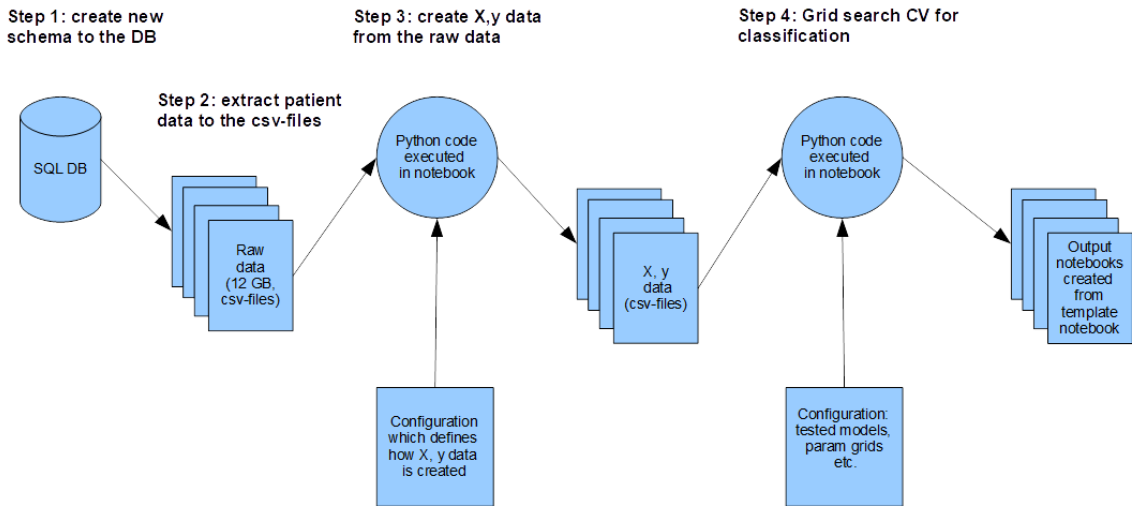

Figure 7. Overview of SW process

This chapter gives a short introduction to software implemented in this study. The SW consists of 23 Python modules that are divided into 6 packages. The implementation includes unit tests for the most important classes. Other implementation includes multiple SQL scripts and Jupyter notebooks. Figure 7 shows an overview of SW process. The aim of the SW implementation was to make it easy to do experiments with different kind of datasets and machine learning models. Most of the experiments were done by repeating dataset creation and grid search CV steps (steps 3 and 4 in Figure 7). Another goal was to store results of the experiments in a format which can be easily accessed later and which allows re-execution of the results when needed. The SW uses template notebooks for configuring both dataset creation and grid search CV. Template notebooks are Jupyter notebooks which are parameterized and executed by Python SW. Template notebook itself is not executed, but output notebooks are created from the template, configured and executed. The output notebook documents the results of the execution and it can also be re-executed later if needed. Template notebook remains unchanged, which makes it easier to separate code from the results when working with Jupyter notebooks. Notebook execution and parameterization was implemented using the Papermill Python library. Project's main program allows combining dataset creation and grid search CV execution for any number of models into one command. Typically these batch jobs lasted a day or two depending on the number of tested models and the size of parameter grids. Final test set evaluation was done by configuring and executing a separate notebook manually. Main steps of data processing in Figure 7 are:

Starting point: SQL DB which contains 6 213 patient stays from Jorvi Hospital's ICU department.

Step 1: Create new schema to the DB which contains intermediate tables, views and functions used to extract patient data from the DB. The functionality is implemented by set of SQL scripts. The schema is created by executing the main script.

Step 2: Extract raw patient data from the newly created DB schema. Data is extracted by executing a SQL script which creates a set of csv files. Size of the created files is about 12 GB.

Step 3: Create new dataset (X, y data) from the raw input data according to configuration. The implementation is Python code which is executed in the Jupyter notebook. The configuration is injected to the Jupyter notebook by project's main script using the Papermill. The notebook creates X, y data and prints statistics about dataset. This step includes e.g. data resampling, imputation, finding valid samples and calculating features and target data from the found samples. The created X, y data is written to csv files that are used to train and test models.

Step 4: Execute grid search cross-validation for the models defined by the configuration. The implementation is Python code which is executed in the Jupyter notebook. The implementation is also based on the template notebook that is parameterized using the Papermill. Project's main script reads the configuration and for each model in the configuration the main script creates a new output notebook from the template notebook, injects model configuration to the output notebook and executes the output notebook. The output of this step is a collection of notebooks in a new directory where each notebook contains grid search CV results for a single model.

5 RESULTS

5.1 Cross-validation results

Model parameter values that provided the best performance were searched using a grid search with 10-fold cross-validation in the development set. Table 11 summarizes tested models, tested parameter values and parameters that provided the best performance. Mean AUROC was used as a criteria for selecting the best parameters. All models were implemented using the scikit-learn machine learning library. Model parameters that are not listed in the Table 11 had scikit-learn default values.

Table 11. Tested hyperparameters and the best parameter values

Model	**Tested parameter values**	**Best parameters (criteria: AUROC)**
LogisticRegression	class_weight: "balanced" penalty: "l1", "l2" C: 0.0001, 0.001, 0.01, 0.1, 1 max_iter: 100000	class_weight: "balanced" penalty: "l1" C: 0.01 max_iter: 100000
SVC	class_weight: "balanced" C: 0.1, 1, 10, 100 gamma: 0.001, 0.01, 0.1, 1	class_weight: "balanced" C: 10 gamma: 0.01
RandomForestClassifier	class_weight: "balanced" max_features: 0.05, 0.1, 0.2, 0.4, 0.6 max_depth: 5, 6, 7, 8, 9 n_estimators: 500, 1000, 2000	class_weight: "balanced" max_features: 0.1 max_depth: 7 n_estimators: 1000
GradientBoostingClassifier	max_features: 0.05, 0.1, 0.2, 0.4 max_depth: 1, 2, 3, 4 learning_rate: 0.001, 0.01, 0.05, 0.1, 0.2 n_estimators: 500, 1000, 2000	max_features: 0.1 max_depth: 2 learning_rate: 0.01 n_estimators: 1000

For SVC, features were scaled using scikit-learn's MinMaxScaler. For other models, data is not scaled. MinMaxScaler scales data so that all values are between 0 and 1. The scaling was implemented using MinMaxScaler as a scikit-learn's pipeline step. MinMaxScaler was also tested with LogisticRegression, but results were slightly better without scaling. Decision tree based models do not need scaling.

Gradient boosting classifier does not have a class_weight argument. For Gradient boosting classifier class weights were calculated using scikit-learn's compute_class_weight-method (see chapter 3.2.5).

Table 12 and Table 13 display cross-validated AUROC and AUPRC values for the models that used the best parameters listed in Table 11.

Table 12. AUROC mean, standard deviation, minimum and maximum from the cross-validation

		AUROC		
Model	**mean**	**std**	**min**	**max**
LogisticRegression	0,797	0,028	0,756	0,844
SVC	0,799	0,027	0,755	0,837
RandomForestClassifier	0,806	0,031	0,745	0,850
GradientBoostingClassifier	0,813	0,029	0,762	0,855

Table 13. AUPRC mean, standard deviation, minimum and maximum from the cross-validation

		AUPRC		
Model	**mean**	**std**	**min**	**max**
LogisticRegression	0,352	0,050	0,289	0,431
SVC	0,355	0,057	0,273	0,440
RandomForestClassifier	0,363	0,068	0,247	0,477
GradientBoostingClassifier	0,375	0,062	0,264	0,457

Gradient boosting classifier was found to be the best model in the cross-validation. Figure 8 displays the learning curve for the gradient boosting classifier which uses the best parameters found in the gid search. The figure shows cross-validated training and test scores as a function of the training set size. The reported score in the figure is AUROC. It can be seen from the figure that when the training set size is in its' maximum, training and test curves are fairly close to each other which means that the figure does not indicate major overfitting. Also, cross-validated test AUROC seems to be reasonably well converged which indicates that this particular model wouldn't significantly benefit from additional data.

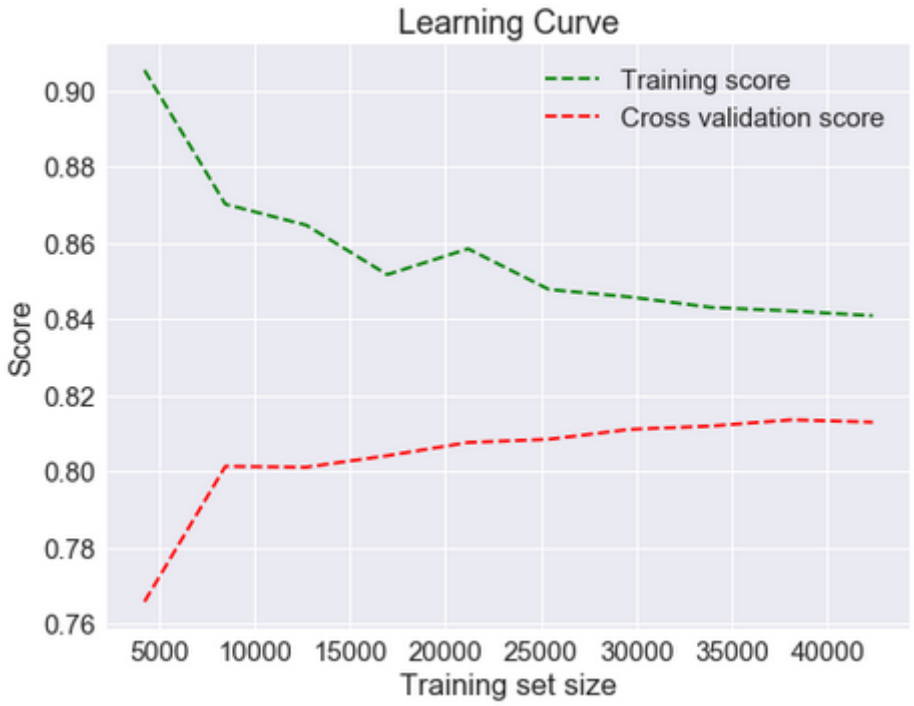

Figure 8. Learning curve of gradient boosting classifier. Score = AUROC.

5.2 Test set results

Gradient boosting classifier was selected to be the best model in the grid search cross-validation. In the final performance evaluation it is trained on the whole development set and evaluated on the test set. Model is initialized with the best parameters found in the grid search cross-validation (Table 11). Model's performance is compared to the baseline performance.

Figure 9 displays the confusion matrix for the gradient boosting classifier from the test set. The confusion matrix is measured using the classifiers's default threshold 0,5. Table 14 displays metrics that are derived from the confusion matrix.

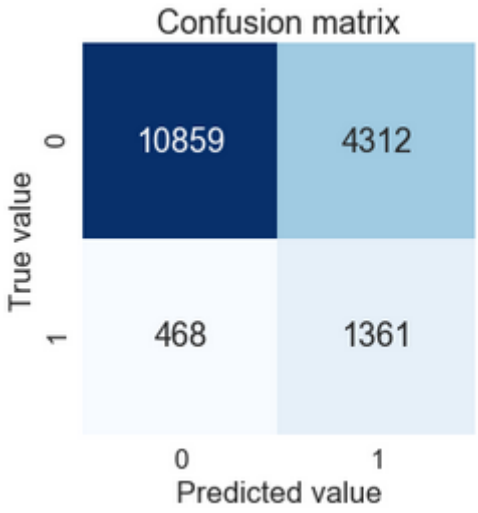

Figure 9. Gradient boosting classifier's confusion matrix from the test set using the default threshold

Table 14. Gradient boosting classifier's test set metrics using the default threshold

Metric	**GradientBoostingClassifier**
Sensitivity, Recall, TPR	0,744
Specificity, TNR	0,716
Precision, PPV	0,240
F1-score	0,363
Positive likelihood ratio (LR+)	2,618
Negative likelihood ratio (LR-)	0,357

Table 15 displays AUROC and AUPRC values for gradient boosting classifier from the test set. Test set AUROC remains almost identical to cross-validated value (0,808 vs 0,813), while AUPRC is slightly lower than in the cross-validation (0,348 vs 0,375). This may be at least partly explained by the prevalence which is slightly lower in the test set (development set:12,8 %, test set: 10,8 %). Figure 10 and Figure 11 display the actual ROC and PR curves. The performance with the model's default threshold 0,5 is marked on both figures.

Table 15. Gradient boosting classifier's AUROC and AUPRC from the test set

Metric	**GradientBoostingClassifier**
AUROC	0,808
AUPRC	0,348

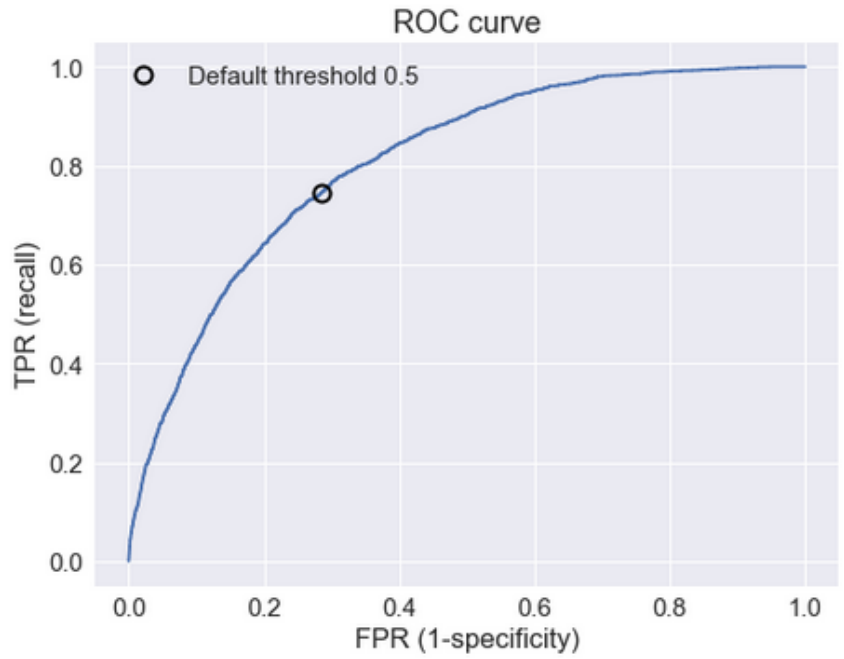

Figure 10. Gradient boosting classifier's ROC curve from the test set

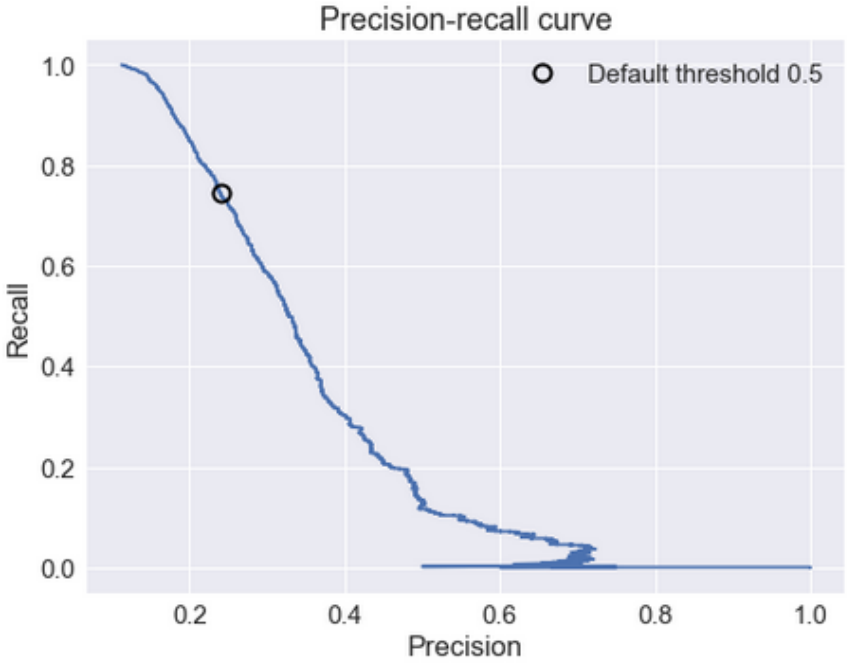

Figure 11.Precision recall curve of gradient boosting classifier from the test set

Figure 12 and Table 16 show the performance comparison between the gradient boosting classifier and the baseline. In order to make the comparison easier, gradient boosting classifier's results are reported using a threshold (0,721) which provides the same sensitivity as baseline. Figure 12 displays confusion matrices for equal sensitivities. Table 16 displays the metrics that are derived from the confusion matrices.

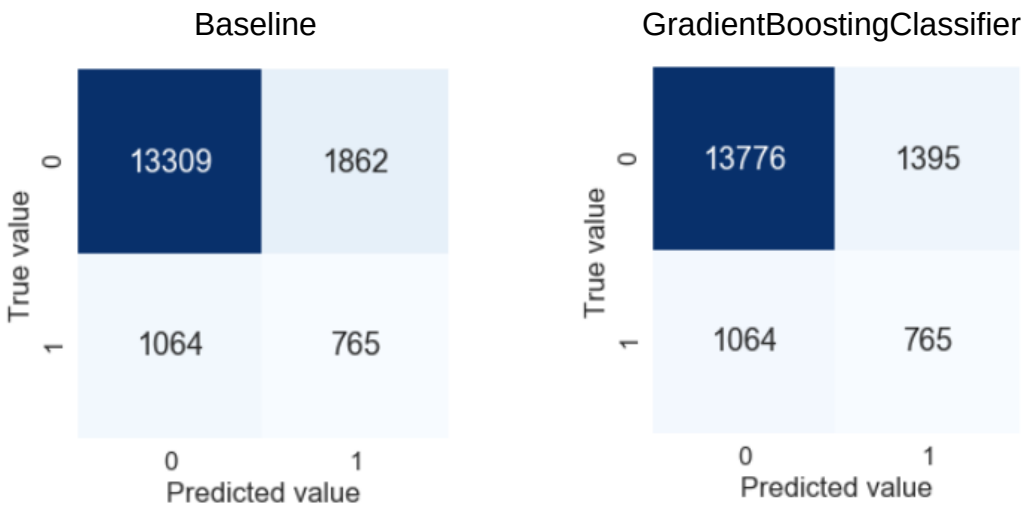

Figure 12. Baseline confusion matrix (left) and gradient boosting classifier confusion matrix (right) from the test set with equal sensitivities.

Table 16. Comparison of test set metrics between the gradient boosting classifier and the baseline. Gradient boosting classifier metrics are reported using a threshold which provides the same sensitivity as baseline.

Metric	**GradientBoostingClassifier**	**Baseline**
Sensitivity, Recall, TPR	**0,418**	**0,418**
Specificity, TNR	0,908	0,877
Precision, PPV	0,354	0,291
F1-score	0,384	0,343
Positive likelihood ratio (LR+)	4,549	3,408
Negative likelihood ratio (LR-)	0,641	0,663

From the confusion matrices of Figure 12, it can be seen that using the same sensitivity, gradient boosting classifier provides 467 less false positives than the baseline which equals to 25 % reduction in false positives. Probability that positive prediction is correct (precision) is 36 % for the gradient boosting classifier and 29 % for the baseline.

Figure 13 and Table 17 show the comparison between the gradient boosting classifier and the baseline using the gradient boosting classifier threshold (0,628) which provides the same precision as baseline.

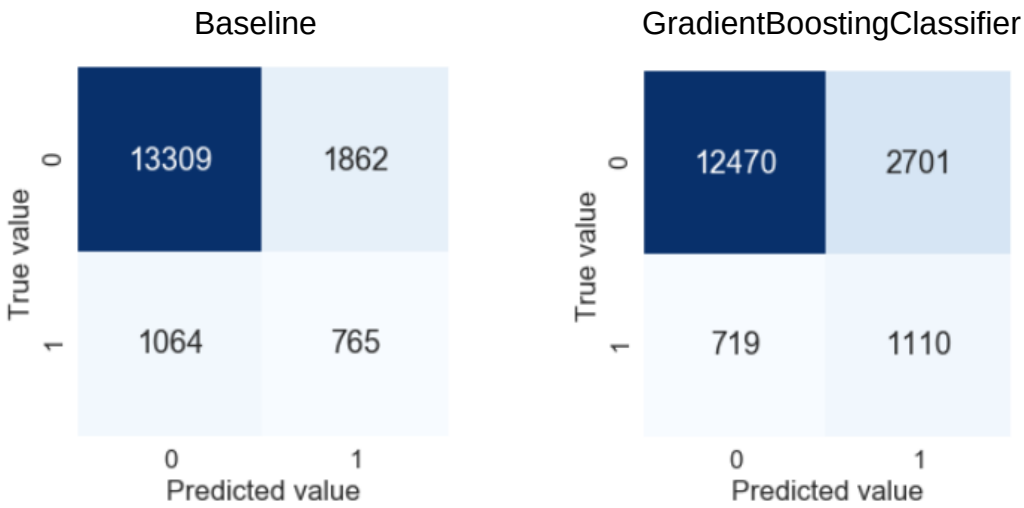

Figure 13 Baseline confusion matrix (left) and gradient boosting classifier confusion matrix (right) from the test set with equal precisions.

Table 17. Comparison of test set metrics between the gradient boosting classifier and the baseline. Gradient boosting classifier metrics are reported using a threshold which provides the same precision as baseline.

Metric	**GradientBoostingClassifier**	**Baseline**
Sensitivity, Recall, TPR	0,607	0,418
Specificity, TNR	0,822	0,877
Precision, PPV	**0,291**	**0,291**
F1-score	0,394	0,343
Positive likelihood ratio (LR+)	3,409	3,408
Negative likelihood ratio (LR-)	0,478	0,663

From the Table 17, it can be seen that using the same precision (0,291) as baseline, the gradient boosting classifier has 45 % higher sensitivity than the baseline. The probability that positive class is predicted positive (sensitivity) is 61 % for the gradient boosting classifier and 42 % for the baseline.

6 DISCUSSION

6.1 Performance

The aim of the study was to predict patient deterioration using the dataset that simulates hospital's general ward which supports continuous monitoring of vital sign parameters. Patient deterioration was defined by National Early Warning Score (NEWS) high-level clinical alert (threshold 7). Four models were tested in binary classification task. Gradient boosting classifier was found to provide the best performance.

While measured test set AUROC of 0,808 indicates good discrimination of positive samples from negative samples, the precision could be improved. The cross-validated AUPRC in the development set was 0,375 (prevalence 12,8%) and in the test set evaluation AUPRC was 0,348 (prevalence 10,8%). Using model's default threshold of 0,5, test set sensitivity was 0,744 and precision was 0,240. Even though the gradient boosting classifier provided 25 % less false positives than the baseline (NEWS medium-level clinical alert), the precision with a default threshold means that there are about three times more false positives than true positives. Alarm fatigue is a well-known problem in hospitals and too many false alarms may even mean that alarms are generally ignored by hospital staff (Bedoya et al. 2019). The precision could be increased by changing the threshold but at the cost of lowering the sensitivity (Figure 11).

The performance could be improved by shortening the gap. If the gap is shortened from one hour to half an hour, gradient boosting classifier's cross-validated AUROC is increased from 0,813 to 0,829 and AUPRC is increased from 0,375 to 0,432. However, the results are now reported using the gap of one hour because it was considered to be more practical. I.e. one hour gap provides more time for the clinical intervention.

Another way of improving the results could be to increase the size of the dataset. Bigger dataset would make it possible to use more complex model without overfitting which might provide better performance. The current dataset which simulates general ward is 64 084 samples that are gathered from 917 patient stays. The dataset is significantly

smaller than the original ICU dataset. A third way of improving the results might be to add new features.

Four different models were evaluated in the cross-validation. Performance difference between the tested models was fairly small but consistent. During the development, several experiments were done where e.g. features, window sizes etc. parameters were varied and gradient boosting classifier consistently provided slightly better cross-validation results than the other tested models. Other benefits of gradient boosting classifier (and also random forest) which makes it a good candidate for different kind of tasks are that it does not require scaling of data, it works well with categorical and continuous data and it provides feature importance information.

6.2 Methods

In ICU, patients are intensively monitored by patient monitoring devices and hospital staff. That's why more patient data is available from the intensive care departments than other hospital departments. The novel idea in this thesis was to select a subset of ICU data to simulate general ward data. Datasets are typically difficult to find, they may be expensive or the wanted dataset may even not exist. The ability to simulate other patient population using e.g. ICU datasets would be a great advantage. However, at the time of the writing of the thesis, it was not possible to evaluate developed model's performance using the real ward data. It remains thus unknown how well the generated dataset simulates the general ward.

An open question related to the simulation is that how well patient's physiological parameters behavior in the simulated data correspond to the real general ward patient's physiological parameters. The process that generates simulated dataset (chapter 3.1.9) can produce several positive samples from the same patient. It means that dataset contains samples from patients that have deteriorated and recovered after that. Patients that have had several deteriorations may have fundamentally different medical state compared to typical patient in the general ward. This issue could be addressed by ignoring data after the first deterioration but it would decrease the size of the dataset too much in this work.

A notable difference between the ICU and general ward is that in the general ward patients can typically move independently which will create more motion artifacts to the data, especially if patients are constantly monitored using wearable wireless sensors. In ICU, patients are more stationary which may mean that the dataset which is used in this study has less noise than real ward dataset. Implementing the same functionality using the real general ward data might require adding a new status feature which tells if the patient is moving or not.

Using a NEWS score to represent patient's medical condition and to determine prediction target can be questioned. NEWS subscores are defined using fixed parameter ranges which do not take into account patient's personal physiological baselines. For example, HR value below 50 may be normal for some people, but NEWS HR subscore is always increased if HR is below this limit. In addition, automatically calculated score does not necessarily always represent well patients real physiological state. For example, SpO2 value is typically measured by a probe in a finger and patient's motion easily causes artifacts in SpO2 value. These artifacts can momentarily raise or lower the score even though patient's real medical condition does not change. A score which is calculated from the measured parameter data is a surrogate of patient's medical condition and it cannot produce as exact labels as manually created labels (e.g. labels for death or cardiac arrest). This thesis tries to address artifacts by calculating 15 minutes median of NEWS which cuts away some short peaks. The benefit of the NEWS is that it catches the deterioration from any clinical reason.

A NEWS value of 7 was used as a threshold for patient deterioration. It is likely difficult for models to separate patients that have e.g. a score of 6 from patients that have a score of 7. A small change in one of the used physiological parameters may shift the sample from negative to positive or vice versa (Table 7). Alternative approach would be to predict the change of NEWS or NEWS value as a regression. These predictions could reveal also those cases where the score changes e.g. from 0 to 6 (Figure 5).

As mentioned earlier, bigger dataset might enable better results. The size of the final dataset could be increased by using bigger ICU dataset as a starting point. Alternatively,

if the same ICU dataset is used, rules for creating the final dataset should be changed. One alternatively would be to change the handling of the temperature parameter. The used temperature parameter significantly decreased the size of the dataset because it was measured in fewer patient stays than other parameters (Table 6). Allowing imputation of temperature parameter or alternatively merging different temperature parameters together would increase the size of the final dataset. Selecting another ICU dataset might also solve this issue.

The size of the dataset affected also the selection of window sizes used in the prediction. In this study, a shorter history window provided better cross-validation results than longer history windows. That is probably because dataset creation rules (chapter 3.1) generate bigger dataset and also higher prevalence for shorter history windows.

In order to be beneficial in practice, the prediction must be actionable. For example, predicting patient deterioration within next 24 hours in ICU may not necessarily be an actionable prediction. In ICU, patients are already in intensive monitoring and if patient does not have clear signs of deterioration, there may not be any practical actions that could be attached to the prediction. Predicting patient deterioration in the general ward, on the other hand, provides actionable predictions. In the general ward, one nurse typically takes care of several patients and knowing which patients have higher risk of deterioration helps hospital staff in directing their resources more efficiently. Patients with higher risk of deterioration can be monitored more intensively by hospital staff.

6.2.1 Alternative ways to define the classification target

This chapter briefly discusses alternative ways to define the prediction target; predicting patient's need for vasopressor administration and predicting patient's need for mechanical ventilation. This information is available in the used dataset and these targets are also used in several examined literature review studies (Fialho et al. 2013; Ghassemi et al. 2017; Suresh et al. 2017; Wu et al. 2017). In hospital, vasopressors are used to raise patient's blood pressure and ventilator assists patient in breathing. These treatments could be used to define the prediction target in addition to NEWS or in place of NEWS. If used in place of NEWS the predicted event would be clinically more

specific compared to the NEWS because NEWS catches the deterioration from any clinical reason.

Table 18 displays dataset statistics for three different targets; patient's need for mechanical ventilation (VENT), patient's need for vasopressor administration (VASOP) and a combination of mechanical ventilation, vasopressor administration and NEWS high level alert. Datasets are created using the rules described by chapter 3.1 except that prediction label is defined differently. I.e. the size of the dataset remains the same but number of positive and negative samples changes. For VENT the sample is positive if mechanical ventilation starts in the prediction window, for VASOP the sample is positive if vasopressor administration starts in the prediction window and for combined target the sample is positive if mechanical ventilation or vasopressor administration starts in the prediction window or max NEWS in prediction window is greater than or equal to 7. A gradient boosting classifier was trained and tested separately for each of the three datasets. The results are left in the discussion because of small number of positive samples for mechanical ventilation and vasopressor administration which causes high variation in results and makes the results less reliable. However, these targets could be useful in future research projects with bigger datasets.

Table 18. Datasets for alternative prediction targets

	Development set	**Test set**
Size (nr of samples)	47 084	17 000
VENT		
Number of positive samples	408	204
Positive samples (percentage)	0,9 %	1,2 %
Nr of patient stays that have positive samples	60	22
Nr of patient stays that have negative samples	672	237
VASOP		
Number of positive samples	273	137
Positive samples (percentage)	0,6 %	0,8 %
Nr of patient stays that have positive samples	37	17
Nr of patient stays that have negative samples	677	234

VENT or VASOP or NEWS>=7		
Number of positive samples	6 416	2 074
Positive samples (percentage)	13,6 %	12,2 %
Nr of patient stays that have positive samples	417	152
Nr of patient stays that have negative samples	637	216

Table 19 and Table 20 show gradient boosting classifier's cross-validation results from the development set. The difference between the min and max AUROC/AUPRC is big for mechanical ventilation and vasopressor administration.

Table 19. Gradient boosting classifier's AUROC from cross-validation for alternative prediction targets

	AUROC			
Target	**mean**	**std**	**min**	**max**
VENT	0,680	0,088	0,535	0,835
VASOP	0,790	0,092	0,626	0,911
VENT or VASOP or NEWS >= 7	0,793	0,032	0,734	0,840

Table 20. Gradient boosting classifier's AUPRC from cross-validation for alternative prediction targets

	AUPRC			
Target	**mean**	**std**	**min**	**max**
VENT	0,026	0,020	0,004	0,060
VASOP	0,034	0,033	0,002	0,109
VENT or VASOP or NEWS >= 7	0,371	0,060	0,264	0,459

Table 21 shows gradient boosting classifier's AUROC and AUPRC from the test set. All AUROCs are between 0,7 and 0,8 indicating reasonable discrimination. Performance for the combined target is slightly worse than for the NEWS alone.

Table 21. Gradient boosting classifier's AUROC and AUPRC from test set for alternative prediction targets

Target	**AUROC**	**AUPRC**
VENT	0,743	0,048
VASOP	0,729	0,024
VENT or VASOP or NEWS >= 7	0,780	0,342

Gradient boosting classifier's most important features indicate that the model is investigating correct information when predicting vasopressor administration and mechanical ventilation. When predicting vasopressor administration, all 10 most important features are derived from blood pressures (SysBP, MeanBP or DiaBP). When predicting mechanical ventilation, 7 out of 10 most important features are derived from respiration rate or SpO2. For the combined target, most important features are more divided between different parameters.

6.3 Recommendations for future research

A question which remains open is how well general ward data can be simulated from the ICU data. This question cannot be answered at the time of the writing of the thesis because real general ward data with continuous vital sign measurements is not available yet. It remains a proposal for future work to test the model performance using real ward data.

Another proposal for future development would be to test the prediction performance with a bigger dataset. With bigger dataset more complex model could be used which might enable improved results. The final dataset could be increased by selecting a bigger ICU dataset as a starting point or by combining several datasets together. Model training data could be combined from different hospitals/countries or from different hospital departments to form a bigger dataset. The performance with alternative prediction targets (chapter 6.2.1) should also be retested with bigger amount of data.

6.4 Limitations

Data that is used in this study is collected from one hospital. Because of differences in hospital practises and patient population, it is quite possible that tested models would not achieve the same performance if they were tested with another dataset. In order to generalize the results, the performance should be tested using data from other hospitals, preferrably also from other countries.

6.5 Conclusion

The aim of the study was to predict patient deterioration from any clinical reason using the dataset that simulates hospital's general ward which supports continuous monitoring of vital sign parameters. NEWS high-level clinical alert (threshold 7) defined the patient deterioration.

The ward data simulation was implemented by selecting a subset of ICU data. The resulted dataset was 64 084 samples from 917 patient stays which was significantly less than the original ICU dataset which contained 6213 patient stays. A bigger dataset would be preferred for the future research. A bigger dataset would give more freedom in selecting prediction window sizes, experimenting with alternative targets and might enable improved results. It was not possible to evaluate how well the generated dataset simulates general ward because real data from general ward was not available yet.

Gradient boosting classifier provided the best performance among the tested models. The achieved AUROC (cross-validation 0,813, test set 0,808) indicates good discrimination of positive samples from negative samples. The achieved precision remains moderate. High number of false positives is typical in the medical domain and it remains a challenge also in this study. Using the model's default threshold, test set sensitivity was 0,744 and precision was 0,240. It means that the model correctly detected 3 out of 4 deteriorations. Among all the predicted deteriorations, the proportion of correct predictions was 1 in 4.

Gradient boosting classifier's performance was compared to NEWS medium level alert. Using the equal sensitivity with the baseline, gradient boosting classifier had 25% less false positives. Using the equal precision with the baseline, gradient boosting classifier had 45% higher sensitivity. The achieved results suggest that real-time prediction of patient deterioration based on the NEWS could assist clinicians in identifying deteriorating patients in hospitals general ward. NEWS is widely used in hospitals general ward and real-time NEWS is already automatically calculated by many hospital systems. Providing NEWS predictions in addition to real-time NEWS could help clinicians in focusing better in most critical patients.

Evaluating how well generated dataset simulates the general ward and testing the prediction performance with a bigger dataset remain suggestions for the future work.

REFERENCES

Alvarez, C. A. et al. (2013) *Predicting out of intensive care unit cardiopulmonary arrest or death using electronic medical record data*, BMC Medical Informatics and Decision Making, 13(1), pp. 1–11.

Bedoya, A. D. et al. (2019) *Minimal Impact of Implemented Early Warning Score and Best Practice Alert for Patient Deterioration*, Critical Care Medicine, 47(1), pp. 49-55.

Calvert, J. S. et al. (2016) *A computational approach to early sepsis detection*, Computers in Biology and Medicine. Elsevier, 74, pp. 69–73.

Cardoso, L. T. Q. et al. (2011) *Impact of delayed admission to intensive care units on mortality of critically ill patients : a cohort study*, Critical Care.

Churpek, M. M. et al. (2012) *Derivation of a cardiac arrest prediction model using ward vital signs*, Critical Care Medicine, 40(7), pp. 2102–2108.

Churpek, M. M. et al. (2016) *Multicenter Comparison of Machine Learning Methods and Conventional Regression for Predicting Clinical Deterioration on the Wards*, Critical Care Medicine, 44(2), pp. 368–374.

Churpek, M. M. et al. (2017) *Quick sepsis-related organ failure assessment, systemic inflammatory response syndrome, and early warning scores for detecting clinical deterioration in infected patients outside the intensive care unit*, American Journal of Respiratory and Critical Care Medicine, 195(7), pp. 906–911.

D'Aragon, F. et al. (2015) *Blood pressure targets for vasopressor therapy: a systematic review*, Shock, 43(6), pp. 530–539.

Decision Trees (2009) scikit-learn 0.19.2 documentation. Available from: https://scikit-learn.org/0.19/modules/tree.html#tree (Accessed 5 November 2019).

Decision tree learning (2019) Wikipedia. Available from: https://en.wikipedia.org/wiki/Decision_tree_learning (Accessed 5 November 2019).

Ensemble methods (2012) scikit-learn 0.19.2 documentation. Available from: https://scikit-learn.org/0.19/modules/ensemble.html (Accessed 6 November 2019).

Fialho, A. S. et al. (2013) *Disease-based Modeling to Predict Fluid Response in Intensive Care Units*, Methods Inf Med, 52(6), pp. 494–502.

Friedman, J. H. (2001) *Greedy function approximation: A gradient boosting machine*, Annals of Statistics, 29(5), pp. 1189–1232.

Generalized Linear Models (2009) scikit-learn 0.19.2 documentation. Available from: https://scikit-learn.org/0.19/modules/linear_model.html#logistic-regression (Accessed 23 November 2019).

Ghassemi, M. et al. (2017) *Predicting intervention onset in the ICU with switching state space models*, Journal of the American Medical Informatics Association, 24(3): pp. 488-495.

Kumar, Anand et al. (2006) *Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock*, Critical Care Medicine, 34(6), pp. 1589–1596.

Le Lagadec, M. D. and Dwyer, T. (2017) *Scoping review: The use of early warning systems for the identification of in-hospital patients at risk of deterioration*, Australian Critical Care. Australian College of Critical Care Nurses Ltd, 30(4), pp. 211–218.

Logistic regression (2019) Wikipedia. Available from: https://en.wikipedia.org/wiki/Logistic_regression (Accessed: 23 November 2019).

Mandrekar, J. N. (2010) *Receiver operating characteristic curve in diagnostic test assessment*, Journal of Thoracic Oncology. International Association for the Study of Lung Cancer, 5(9), pp. 1315–1316.

Model evaluation: quantifying the quality of predictions (2010) scikit-learn 0.19.2 documentation. Available from: https://scikit-learn.org/0.19/modules/model_evaluation.html#log-loss (Accessed 30 November 2019)

Müller, A. C. & Guido S. (2016) *Introduction to Machine Learning with Python*, O'Reilly Media, Third Release, 378 pages.

Parr, T. & Howard, J. *How to explain gradient boosting*. Available from: https://explained.ai/gradient-boosting/ (Accessed 29 September 2019).

Pearse, R. M. et al. (2012) *Mortality after surgery in Europe : a 7 day cohort study*, The Lancet. Elsevier Ltd, 380(9847), pp. 1059–1065.

Positive and negative predictive values (2019) Wikipedia. Available from: https://en.wikipedia.org/wiki/Positive_and_negative_predictive_values (Accessed 30 October 2019).

RandomForestClassifier (2017) scikit-learn 0.19.2 documentation. Available from: https://scikit-

learn.org/0.19/modules/generated/sklearn.ensemble.RandomForestClassifier.html (Accessed 31 October 2019).

Royal College of Physicians. *National Early Warning Score (NEWS) 2: Standardising the assessment of acute-illness severity in the NHS. Updated report of a working party*. London: RCP, 2017.

Smith G. B. et al. (2013) *The ability of the National Early Warning Score (NEWS) to discriminate patients at risk of early cardiac arrest, unanticipated intensive care unit admission, and death*, Resuscitation, 84(4), pp. 465–70.

Support Vector Machines (2017) scikit-learn 0.19.2 documentation. Available from: https://scikit-learn.org/0.19/modules/svm.html#svm-kernels (Accessed 8 December 2019).

Suresh H. et al. (2017) *Clinical Intervention Prediction and Understanding using Deep Networks,* arXiv:1705.08498.

Torsvik, M. et al. (2016) *Early identification of sepsis in hospital inpatients by ward nurses increases 30-day survival*, Critical Care, 20.244, pp. 1–9.

VanderPlas, J. (2017) *Python Data Science Handbook*, O'Reilly Media, First Release, 529 pages.

Wu, M. et al. (2017) *Understanding vasopressor intervention and weaning: risk prediction in a public heterogeneous clinical time series database,* Journal of the American Medical Informatics Association, 24(3), pp. 488–495.